Probabilistic Approaches to Consumer-generated Review Recommendation

Richong Zhang

Thesis Submitted to the Faculty of Graduate and Postdoctoral Studies

In partial fulfilment of the requirements for the degree of

Doctor of Philosophy in Computer Science

School of Information Technology and Engineering

Faculty of Engineering

University of Ottawa

Abstract

Consumer-generated reviews play an important role in online purchase decisions for many consumers. However, the quality and helpfulness of online reviews varies significantly. In addition, the helpfulness of different consumer-generated reviews is not disclosed to consumers unless they carefully analyze the overwhelming number of available contents. Therefore, it is of vital importance to develop predictive models that can evaluate online product reviews efficiently and then display the most useful reviews to consumers, in order to assist them in making purchase decisions.

This thesis examines the problem of building computational models for predicting whether a consumer-generated review is helpful based on consumers' online votes on other reviews (where a consumer's vote on a review is either HELPFUL or UNHELPFUL), with the aim of suggesting the most suitable products and vendors to consumers. In particular, we propose in this thesis three different helpfulness prediction approaches for consumer-generated reviews. Our entropy-based approach is relatively simple and suitable for applications requiring simple recommendation engine with fully-voted reviews. However, our entropy-based approach, as well as the

existing approaches, lack a general framework and are all limited to utilizing fully-voted reviews. We therefore present a probabilistic helpfulness prediction framework to overcome these limitations. To demonstrate the versatility and flexibility of this framework, we propose an EM-based model and a logistic regression-based model. We show that the EM-based model can utilize reviews voted by a very small number of voters as the training set, and the logistic regression-based model is suitable for real-time helpfulness predicting of consumer-generated reviews. To our best knowledge, this is the first framework for modeling review helpfulness and measuring the goodness of models. Although this thesis primarily considers the problem of review helpfulness prediction, the presented probabilistic methodologies are, in general, applicable for developing recommender systems that make recommendation based on other forms of user-generated contents.

Acknowledgements

I am indebted to my advisor Dr. Thomas Tran for the support he provided in this research. In particular, I am grateful for the time and effort he has spent to discuss challenging problems we faced in the research and the criticism as well as the encouragement he has given me over the last five years. My thesis would not be completed without his supervision. I am also grateful to Dr. Yongyi Mao who kindly provided many useful tips, suggestions and helpful discussions that led the substantial improvements in the thesis. He has also led me to see things in a better light and a clear point of view. I would also like to thank University of Ottawa for providing a wonderful educational environment and financial support for attending conferences.

More thanks go to all my friends for keeping me positive and helping me get through these years. Finally, I would like to give special thanks go to my parents for their support and encouragement during the years of study. I owe all my success to their unconditional love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we discuss the motivation and background of this research. This chapter also presents the problems that is solved in this thesis, the contributions of our research, and the organization of this thesis.

1.1 Motivation

The evolution of the Internet over the past decade has positioned it to be the largest-ever platform of interaction for its users. A user accesses the Internet not only for the purpose of acquiring information but also for the purpose of interacting with other users and for providing information. Especially in the online shopping environment, consumers would like to comprehensively investigate the reputation of the product they are interested in when making purchasing decisions. Consumer-generated re-

views are playing an increasingly significant role and studies have continuously shown the following:

- The quantity and quality of consumer-generated reviews for specific products have had a positive effect on purchaser's intentions [Park et al., 2007; Park and Kim, 2008; Zhu and Zhang, 2010].

- "Online product reviews provided by consumers who previously purchased products have become a major information source for consumers and marketers regarding product quality" [Hu et al., 2008].

- Consumers seek consumer-generated reviews as *Information Input* to help them make purchase decisions; consumer-generated reviews could help a consumer to decide whether a product should be purchased [Robert M. Schindler, 2005; Park and Lee, 2008; Zhang et al., 2010].

In fact, consumer-generated reviews are collectively considered a rich source of information and are increasingly emerging as a *new genre* all on their own [Pollach, 2006]. According to a survey of 1,000 online shoppers [Palmer, 2009], 47 percent of consumers surveyed rely on online consumer reviews when making a purchasing decision. In another survey [Nielsen-Online, 2008], 71 percent of consumers agree that consumer reviews make them more confident that they are buying the right product. Furthermore, 81 percent of online holiday shoppers read online customer reviews according to [Nielsen-Online, 2008].

Moreover, with the development of Web 2.0, which emphasizes the communication and participation, consumers are offered tremendous opportunities to publish their own opinions and experiences. These consumer-generated reviews are found most frequently on electronic commerce web sites like Amazon.com that provide not only products and services but also consumer-generated reviews that are written by previous purchasers. For example, Ebay.com members are always asked to leave comments on a seller after a transaction. Amazon.com provides a platform whereby someone can review products. Product review aggregation web sites such as Epinion.com, Pricegrabber.com, ResellerRatings.com, BizRate.com and Tripadvisor.com each provide consumers with platforms where they can exchange their opinions about products, services and merchants. Digg.com allows users to comment on and rate online articles whereas Tripadvisor.com, as a travel directory, supports reviewers sharing their experiences with other travelers. Users in these communities can share their reviews of any product and any store online.

By taking advantages of the increasing availability of rich information, consumers are assisted by an increased number of user published experiences more than ever before. Still, the quantity of the consumer-generated reviews is immense. Especially with the explosive growth of the number of web blogs, forum posts and consumer-generated reviews, potential consumers have to spend an enormous amount of time retrieving reviews that can assist them in better understanding their intended products.

However, consumer-generated reviews, most of which are written anonymously, vary in quality. This is largely attributed to the heterogeneous nature of the Internet population, where users are from different backgrounds and have different biases. This presents difficulties for consumers to assess the reviews, and such difficulties are greatly emphasized as the number of consumer-generated reviews available on E-Commerce sites and online forums have been increased by many orders of magnitude.

To ease this burden for consumers when searching for reviews that may prove helpful for their purchasing decisions making, a system that can discover the most helpful consumer-generated reviews is needed to reduce the time, effort and difficulties associated with acquiring helpful reviews.

1.2 Research Problem

There is an enormous amount of information contained in consumer-generated reviews which makes building recommender systems based on these reviews very appealing. Indeed, significant research efforts have been carried out over the past years along this direction (see e.g. [Adomavicius and Tuzhilin, 2005] for a comprehensive review of this area). Most of the existing recommendation approaches [Goldberg et al., 1992; Sarwar et al., 2001; Resnick et al., 1994] are based on star ratings of the products. Using a star rating scale, users cannot determine the "real semantics" of a review contents. Since product reviews represent reviewers' feelings, experiences and opinions on a specific

product, they are more useful than product ratings and therefore are better positioned to help potential consumers make purchase decisions. Nevertheless, product reviews are unstructured and often a mix of between reviewer feelings and opinions. Therefore, an effective model to discover the most helpful reviews is essential.

While search engines are good tools to assist in looking for information, the results returned by a search engine are massive. If an individual were to input 'xbox 360 reviews' into Google, there would be 47,100,000 web pages returned. This massive load of information is undoubtedly difficult for any user to handle. Additionally, an online community like Epinion.com usually receives more than 1,000 reviews submitted by different users for a specific product. This justifies the need to develop systems that can recommend helpful reviews to consumers in an effective fashion. We notice that most of the review aggregation web sites provide helpfulness voting functions for consumers to rate reviews. That is, a consumer can vote a particular product review to be "Helpful" or "Not Helpful" after he/she has read the review.

Figure 1.1 shows an online review page from Amazon.com. As you can see from the figure, the product reviews are sorted by "most helpful first" and "335 of 339 people" found that the first review was helpful. In total there are 39 pages of reviews concerning this product and all readers are asked to vote for the review by answering the question: "Was this review helpful to you?"

Potential consumers may make use of these voter opinions ("Helpful" or "Not Helpful") to estimate the helpfulness of the review and to decide whether to read it

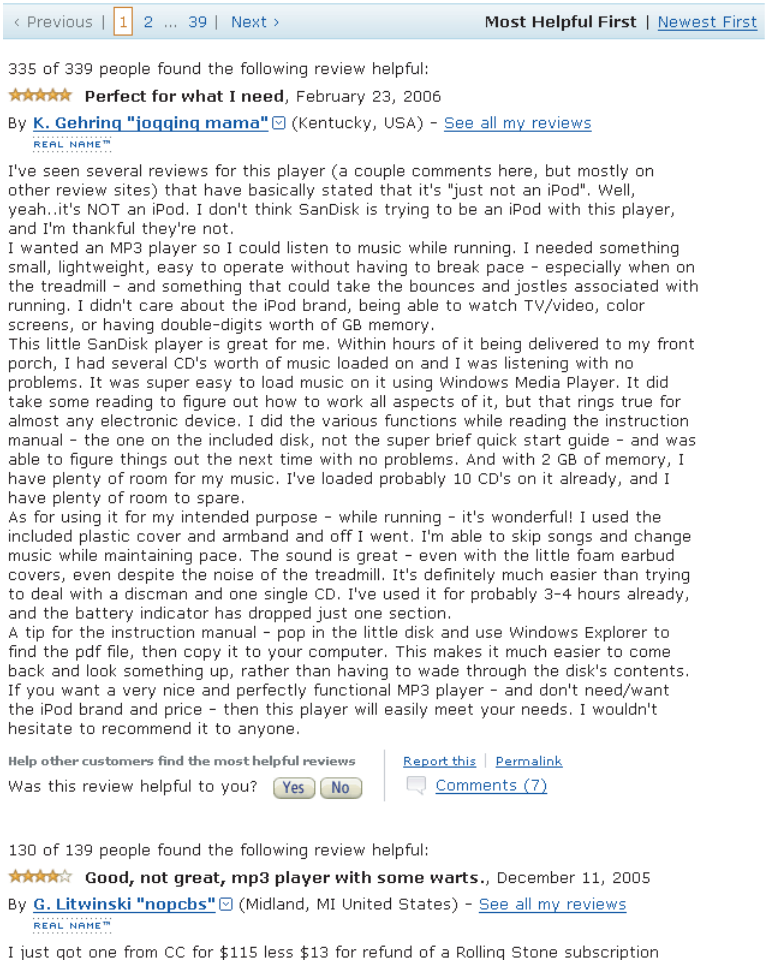


Figure 1.1: An online review page

or not. Nevertheless, the accumulation of voter opinions takes time before an actual helpful review can be discovered. Moreover, the latest published review will always have the least amount of votes regardless of its quality. To address these problems, our goal is to develop models that can effectively filter out the most likely helpful reviews for consumers, in order to provide more valuable information for their decision making process.

In general, the problem of developing a recommender system from consumer-

generated reviews may be abstracted in terms of a collection of *"items"*, a collection of *users*, and each user's *opinion* on a subset of the items. In this case a "item" can refer to a movie, a video clip, a product, a blog or an article; and the "opinion" of a user on a item can be in text form (such as an review article), numerical form (such as a product rating), categorical form (such as tags) or in binary form (such as `LIKE/DISLIKE`). The objective for developing the recommender system may include deciding which items are to be recommended to a particular user or to a typical user, and deciding what level of recommendation should be given among other things.

A typical example of a recommender system that we will consider throughout this thesis is a "review helpfulness predictor" where each "item" is a consumer-generated review of a product and the user opinions are in binary forms, that is, each user having read a review may vote the review as being `HELPFUL` or `UNHELPFUL`. We will consider the case where there is no information about who votes on which review; [1] that is, for each review, in addition to its text content, the only information available is the number of positive (i.e. `HELPFUL`) votes and the number of negative (i.e. `UNHELPFUL`) votes. The functionality of a review helpfulness predictor is to predict the "helpfulness" of a new review (namely, a review that has not received any vote) based on the existing reviews and the existing votes on those reviews. A good review helpfulness predictor is an important component of an E-commerce web site since a large and increasing quantity of Internet users now base their purchase decisions on online product reviews and the ability to find useful reviews rapidly is in great demand.

Compared with the "official" sources of information, e.g. expert and professional reviews, consumer-generated review or user opinions have their unique characteristics.

[1] Having the additional information available about who has voted on which review is a conceptually simpler case, although more sophisticated techniques are required to exploit such information.

Albeit the richness of their information content, the opinions provided by individual users are typically less formal, often biased, and have relatively low reliability with large quality variation. As such, retrieving information from such sources is often a challenging task.

There exists some helpfulness assessment approaches [Kim et al., 2006; Weimer et al., 2007; Weimer, 2007; Liu et al., 2008] which apply general machine learning tools to infer the helpfulness of online reviews. However, these works all lack transparency in their algorithmic behavior, they merely apply traditional non-probabilistic machine learning models and they ignore the uncertainty about the helpfulness estimate. The output of these approaches is a deterministic value and has no probabilistic meaning. Furthermore, no previous work has been completed with which to build a generalized framework to model and evaluate the helpfulness of reviews. There also exist other issues for the helpfulness predicting problem, such as the "words of few mouths" phenomenon and the real-time updating the helpfulness model as new reviews and votes become available (see discussions in the following sections).

1.2.1 "Words of Few Mouths" Phenomenon

The challenge for discovering the helpfulness of consumer-generated reviews is often amplified by a phenomenon called "words of few mouths". Here, "words of few mouths" is a term we coin in this thesis and it refers to the phenomenon where there is a large fraction of "items" each only having received feedbacks from very few users. For example, Figure 1.2 plots the histogram of the number of votes on 9955 reviews at Amazon.com where more than 50% of the reviews have votes by no more than 5 users. The "words of few mouths" phenomenon at Amazon.com shown in this figure is not an isolated case. In fact, such a phenomenon widely exists on many E-commerce web sites. The underlying reason for this phenomenon is perhaps that except for a small number of "popular" ones, most "items" are "unpopular".

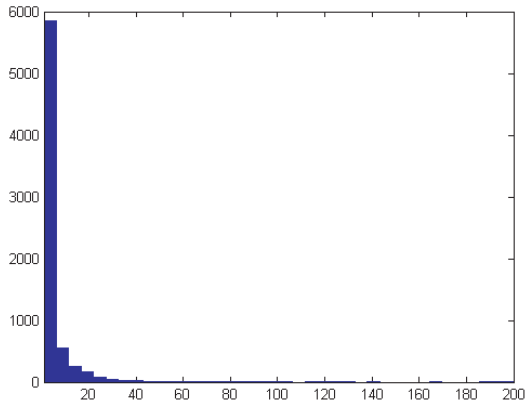


Figure 1.2: Number of votes in the review document data set. Y axis represents the number of review documents receiving a certain number of votes on x-axis (0-5, 6-10, etc.). X-axis: the number of votes a review document received.

The challenges brought by the "words of few mouths" phenomenon in the development of a recommender system manifest itself as further degraded reliability of user feedbacks. Concerning the review helpfulness prediction problem, when each review has been voted by a large number of users, the fraction of positive (i.e. `HELPFUL`) votes is a natural indicator of the "helpfulness" of the review and one can use such a metric to train a learning machine and to infer the dependency of positive vote fractions on review documents (see, e.g., [Kim et al., 2006], [Weimer, 2007] and [Liu et al., 2008].). However, in the case where most reviews are voted exclusively by only a few users, the positive vote fraction is a poor indicator of the review helpfulness and the performance of a predictor trained this way necessarily degrades. We argue that the helpfulness of a review should depend on the review document itself and on the *statistics* of the reader population, unless there is a large number of votes attached to the review document. Otherwise the positive vote fractions is a poor indication of the statistics of the reader population. For example, using positive vote fraction as helpfulness metric, the fractions $\frac{1}{3}$, $\frac{9}{27}$, and $\frac{30}{90}$ are all equal, but one can hardly reason that the corresponding review documents are equally helpful.

In general, the negative impact of the "words of few mouths" problem can have varying severity depending on whether there is additional information available, the size of the data set, the heterogeneity of the "items" and that of "users", and so on. A partial cure for the "words of few mouths" problem is to remove the "unpopular items" from the data set when developing a recommender system. Such an approach is however often unaffordable, particularly when the problem space is large and the data set is relatively small.

1.2.2 Online Prediction of Helpfulness

In practice, consumer-generated reviews and voter opinions accumulate in a dynamic fashion. As more new consumer-generated reviews continuously become available, there is a need to improve the model that handles incoming data. The computational complexity of existing approaches renders impossible the applicability of a real-time system for these existing approaches. Online algorithm is a model can process incoming data one-by-one without requiring entirely training data set is available. Off-line algorithms can only achieve acceptable accuracy when a large number of voter opinions are given. However, online algorithms can begin processing with even a small number of vote opinions and incrementally update the model parameters as more consumer-generated reviews and votes are added to the system. To the best of our knowledge, the online algorithm has not been applied to the recommender systems for consumer-generated reviews.

1.3 Research Methodology

This thesis advocates probabilistic approaches to developing recommender system from consumer-generated reviews, where we focus on the review helpfulness prediction problem.

A key advantage of the probabilistic modeling methodology is that it describes problems in the most concise way. A probabilistic model provides greater insight and perspective about real-world issues; the probabilistic inference gives a theoretical and mathematical description of the model. Furthermore, a probabilistic model outputs a probability that indicates how likely an event will happen. In this thesis we propose three novel probabilistic models with solid mathematical foundations to solve the helpfulness discovery problem. The first model, a procedural entropy-based approach, is employed as a first step to discover the helpfulness of reviews and to rank review documents. This model has been verified with excellent time complexity and the effectiveness has been empirically demonstrated. This modeling approach is a procedural approach that is simple and efficient and the model can be developed rapidly. However, this model mixes the decision rules with the features. As new features come in and the system grows, the model has to be rebuilt for the new task. Also, while solving this model we often find the traditional positive vote fraction based helpfulness metric is fundamentally limited, particularly when only a limited number of votes are available.

We then investigate the "words of few mouths" phenomenon and propose the second model, a declarative review recommendation framework for inferring the helpfulness distribution of consumer-generated reviews using the distribution of helpfulness conditioned on the features inherent to the document to characterize the dependency of the helpfulness and the review. Also, a new helpfulness metric for evaluating the helpfulness of consumer-generated reviews is given in this framework. We demonstrate that this benchmark is more suitable and more reliable than the positive vote fraction for tasks of helpfulness discovery. Unlike most existing helpfulness assessment approaches, we propose a review recommendation framework using the probability density of the review helpfulness as the benchmark. The probability density is the a posterior distribution of the helpfulness value of the review document given the voters' opinions. This naturally accounts for small numbers of votes on a review

document, a scenario in fact dominates the statistics of most review aggregation web sites.

This proposed framework further exploits a probabilistic graphical model as a generative model of voters' opinions and includes an Expectation Maximization (EM) algorithm for the inference of review helpfulness. Experimental results demonstrate that the proposed framework and algorithm are superior to existing approaches.

Finally, we study the problem of designing online probabilistic helpfulness models based on existing consumer-generated reviews and the reader' opinions (helpful or unhelpful votes). Building under our proposed framework, we first develop logistic regression based probabilistic model and off-line learning algorithm. We experimentally compare the logistic regression method with the Support Vector Regression (SVR) method, the current state of the art for such applications, and demonstrate the superior performance of the logistic regression method. We then extend the logistic regression based model to an online algorithm to incrementally update the parameters of the model. Finally, an efficient hybrid algorithm by combining the online and offline alogrithm is provided to increase the convergence rate and prediction precisions. The final two algorithms are tested on real-life consumer-generated reviews and experimental results illustrate that the hybrid approach efficiently processes incoming data and generates a reliable helpfulness prediction for users.

1.4 Contribution

The contributions of this work are as follows:

1. An entropy-based approach (Chapter 3) for modeling the helpfulness of online product reviews is delivered. The time complexity for this model is shown significantly less than other existing approaches.

2. The "words of few mouths" phenomenon (Chapter 4.1) is investigated and a rig-

orous probabilistic framework for inferring the "helpfulness" of customer reviews is proposed. Under this framework, the "helpfulness" of a review document is given a precise mathematical meaning.

3. An evaluation methodology (Chapter 4.2), which utilizes the log-likelihood of the helpfulness prediction model built under the proposed framework, is established for evaluating the overall quality of the helpfulness estimates and the helpfulness inference algorithms are carried out. Also, this framework is not limited to a low number of votes on a review document, a situation that can hardly be handled by existing models.

4. A probabilistic graphical model (Chapter 4.3) and a simple and computationally-efficient Expectation Maximization algorithm (Chapter 4.4) for helpfulness inference are proposed to estimate voter opinion distribution.

5. A logistic regression-based probabilistic model and learning algorithms (Chapter 5) under the proposed framework that incrementally updates parameters as newly generated reviews become available is presented. Based on this model three learning algorithms (Batch, Online and Hybrid) are developed.

Although this thesis primarily considers the problem of customer-generated review helpfulness prediction, the presented probabilistic methodology is in general applicable for developing other recommender systems based on user-generated contents or electronic word of mouth.

1.5 Peer reviewed publications

Different parts of this work have been published or accepted for publication as six journal papers and seven conference papers.

1.5.1 Papers in Refereed Journals

1. Richong Zhang, Thomas Tran, Yongyi Mao. Real-time Helpfulness Prediction based on Voter Opinions. Accepted for publication in Concurrency and Computation: Practice and Experiment, Wiley. 15 pages, Accepted, April 2011.

2. Richong Zhang, Thomas Tran, Yongyi Mao. Opinion Helpfulness Prediction in the Presence of "Words of Few Mouths". Accepted for publication in World Wide Web: Internet and Web Information Systems, Springer. 19 pages, Accepted, March 2011.

3. Richong Zhang and Thomas Tran. A Helpfulness Modeling Framework for Electronic Word-of-Mouth on Consumer-Opinion Platforms. ACM Transaction on Intelligent Systems and Technology, vol. 2, no. 3, April 2011.

4. Richong Zhang and Thomas Tran. A Linear Summarization Approach for Exploiting Product Reviews to Help Online Shoppers Make Good Decisions. Journal of Electronic Commerce Research, vol. 11, no. 3, pp. 220-230.

5. Richong Zhang and Thomas Tran. An Information Gain Based Approach for Recommending Useful Product Reviews. Knowledge and Information Systems, Springer, vol. 26, no. 3, pp. 419-434.

6. Richong Zhang and Thomas Tran. Automatically Filtering Useful User-Generated Contents Using N-Gram Features. IADIS International Journal of WWW /Internet, vol. 7, no. 2, pp. 109-121.

1.5.2 Papers in Refereed Conferences

7. Richong Zhang, Thomas Tran, and Yongyi Mao. Recommender Systems from "Words of Few Mouths". Accepted for publication in Proceedings of the 22nd

International Joint Conference on Artificial Intelligence (IJCAI-2011), 6 pages, Accepted, April 2011.

8. Richong Zhang and Thomas Tran. Probabilistic Modeling of User-Generated Reviews. In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-2010), March 2010 (Acceptance Rate: 22.7%).

9. Richong Zhang and Thomas Tran. A Novel Approach for Recommending Ranked User-Generated Reviews. In Proceedings of the 23rd Canadian Conference on Artificial Intelligence (AI-2010), May 2010.

10. Richong Zhang and Thomas Tran. Review Recommendation with Graphical Model and EM Algorithm. In Proceedings of the 19th International World Wide Web Conference (WWW-2010), April 2010 (Acceptance Rate: 14%).

11. Richong Zhang and Thomas Tran. Helping E-Commerce Consumers Make Good Purchase Decisions: A User Reviews-Based Approach. In Proceedings of the 4th International Conference on E-Technologies (MCETECH-2009), May 2009.

12. Richong Zhang and Thomas Tran. An Entropy-Based Model for Discovering the Usefulness of Online Product Reviews. In Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence (WI-2008), December 2008 (Acceptance Rate: 20%)

13. Richong Zhang and Thomas Tran. An Approach for Predicting and Ranking Consumer Review Helpfulness. In Proceedings of the 2008 IADIS International Conference on E-Commerce (IADIS E-Commerce-2008), July 2008 (Outstanding Paper Award, Acceptance Rate: 23.5%).

1.6 Organization of Thesis

This thesis is divided to three parts. Part 1 (Chapter 2) outlines the related works. Part 2 (Chapters 3, 4, and 5) present the main contribution of this thesis, which is to theoretically model the helpfulness of customer-generated reviews and to propose three models for different scenarios. Part 3 (Chapter 6) then concludes the thesis and discusses the possible future directions for research. A more detailed description is given below:

- Chapter 2 discusses existing works related to the helpfulness discovery problem.
- Chapter 3 delivers an entropy-based approach for modeling the helpfulness of online product reviews.
- Chapter 4 introduces a declarative probabilistic framework with which to estimate the helpfulness distribution of a review and to evaluate the performance of a helpfulness discovering model. We also propose a probabilistic helpfulness metric for the conventional machine learning based predictors. We then develop, under this framework, a probabilistic graphical model and an Expectation Maximization algorithm for helpfulness inference. The experimental results obtained by this model show that our approach gives stat-of-the-art effectiveness.
- Chapter 5 proposes a discriminative online model under the framework proposed in Chapter 4 for handling the real-time helpfulness discovery problem. Basically, this model is a logistic regression based probabilistic learning algorithm. We then extend this model to a hybrid approach by combining the online and batch algorithm. Finally, we experimentally show the effectiveness of the proposed model on a simulated real-time consumer-generated review accumulating process.

- Chapter 6 discusses our future research directions on feature selection, evaluation metrics, modeling algorithms and other issues left unaddressed by our framework.

Chapter 2

Related Work

In this chapter we introduce some related works and fundamental techniques used in this thesis. Section 2.1 briefly surveyes technologies used in recommender systems. Section 2.2 discusses the effect of consumer-generated reviews for consumers and merchants. Section 2.3 shows existing approaches for discovering useful information from consumer-generated reviews. Section 2.5 introduces the state of the art probabilistic modeling approaches in text mining domain, which is the modeling approach used in this work.

2.1 Recommender System

In order to generate appropriate recommendations and to ensure the performance of recommendation systems, researchers have proposed different approaches such as collaborative filtering [Huang et al., 2007; Cheung and Tian, 2004] and content-based [Popescul et al., 2001; Mooney and Roy, 1999] recommendation techniques. In addition, some model-based recommendation systems, like those making use of Bayesian networks [Breese et al., 1998; Huang and Bian, 2009], are also proposed. Data mining technologies, e.g., clustering [Nnadi, 2009; Demir et al., 2007] and association rules

[Lin et al., 2002; Castagnos et al., 2008], are introduced to build recommender systems as well.

A content-based recommender system generates recommendations based on the content of items rather than a user's opinions on these items. Within this system, items are viewed as consisting of features. This method calculates the correlation between features and finds the most relative items to recommend to users. The basic idea of the content-based method is that users prefer items similar to the ones they have previously purchased. In general, content-based recommender systems are suitable for recommending text documents but are not able to find similar implicit features like interests and tastes. This method is not sufficient for determining a user's potential interests.

Collaborative filtering recommender system [Goldberg et al., 1992] predicts the overall ratings by aggregating the experience of other users who are similar to the current user with respect to their interests or other aspects. This type of recommender system, such as GroupLens [Resnick et al., 1994], works based on user ratings and can be used to generate recommendations about movies, music or news.

Sarwar et al. [2001] suggests that item-based collaborative filtering algorithms can perform many recommendations for millions of users and items in seconds, and the Mean Absolute Error generated by item-based collaborative filtering algorithms is lower than that generated by user-based algorithms. This indicates that item-based algorithms are able to provide higher quality recommendations.

Hybrid recommender systems are also proposed in the literature. An example of this is Fab [Balabanovic, 1997] that combines both content-based and collaborative filtering techniques to recommend documents. Combining these two techniques can overcome the disadvantages of using each technique alone and can increase the performance of the system. Furthermore, model-based collaborative filtering is used to establish a model from user behaviors and to generate recommendations based on this model.

The main intention of the above research is to generate recommendations to consumers based on different underlying approaches. However, the potential consumers do not have a chance to clearly understand why they received such recommendations, nor do they have good confidence in following them. In many circumstances, consumers want to hear from other people who have used the products that they are now interested in purchasing.

2.2 The Effect of Consumer-Generated Reviews

A large body of literature investigates the effect of online product reviews on the product sales and consumer behavior. Park et al. [2007] find that the quality of reviews has a positive effect on product sales as consumer purchase intentions increase with the quantity of product reviews. Hu et al. [2008] point out that consumers not only consider review ratings, but also the contextual information like a reviewer's reputation. They also find that the impact of online reviews in sales diminishes over time.

Park and Kim [2008] investigate the relationship between different types of reviews and consumers. They find that consumer concerns vary at each stage of the product life cycle and they suggest that marketers develop different strategies for different types of consumers. Lee et al. [2008] examine the effect of the quality of negative online reviews on product attitude and discovered that high-involvement consumers consider the quality of negative reviews and low-involvement consumers tend to conform to other reviewer attitudes regardless of the quality.

In [Vermeulen and Seegers, 2009], the authors study the effect of online hotel reviews on consumer consideration and concluded that positive reviews have a positive impact on consumer behavior. Park and Lee [2008] analyze the two roles of the online consumer reviews (an informant and a recommender). When information overload occurs, they found that low-involvement consumers mainly focus on the perceived

popularity and high-involvement consumers focus on the product information over reviews.

Based on the findings of these researches, it can be concluded that consumer-generated reviews plays an important role in the purchase decision-making process of consumers as consumers are willing to consider previous consumers' experiences before they decide to make a purchase.

2.3 Text Analysis on Consumer-Generated Contents

Some researchers have been working on sentiment classification, or polarity classification, to predict whether the opinions expressed are positive or negative. Das and Chen [2007] process messages on the Yahoo message board to analyze the opinion of investors about the stocks. Dave et al. [2003] apply classification algorithms on different feature sets for automatically distinguishing positive and negative reviews. Hatzivassiloglou and McKeown [1997] propose a method to predict the positive or negative semantic orientation of adjectives based on a supervised learning algorithm. They introduce a log-linear regression to predict the conjoined adjectives' orientation. A clustering algorithm is also introduced to group adjectives in a positive or negative class. Turney presents an unsupervised learning algorithm to classify reviews as being recommended or not recommended by analyzing their semantic orientation based on mutual information. The average semantic orientation of the phrases of a reviews is calculated and the label of the review is determined by this average semantic orientation. In this approach, the semantic orientation of the phrases is calculated by the difference between the mutual information of the positive words and the negative words. Yu and Hatzivassiloglou [2003] propose a classification approach to retrieve opinion sentences and to separate these opinion sentences as positive or negative.

In [Pang et al., 2002], the authors classify movie reviews as positive or negative by utilizing several machine learning methods, namely Naive Bayes, Maximum Entropy and Support Vector Machines (SVM). They also make use of different features like unigram, bigram, position and the combination of these features. Their results show that the unigram presence feature is the most effective and the SVM performs the best for sentiment classification.

Another research domain related to consumer-generated reviews is review mining and summarizing. In [Zhuang et al., 2006] the authors mine and summarize the movie reviews based on a multi-knowledge approach which includes WordNet, statistical analysis and movie knowledge. Hu and Liu [2004] summarize product reviews by mining opinion features. Wong and Lam [2008] propose an approach to summarize item features and properties from multiple websites. Inui et al. [2008] suggest a model for collecting instances of personal experiences as well as opinions from user-generated content. Under this system consumers can perform searches for the experiences and opinions of others related to one or more topics. In addition, the experiences returned from the system can be automatically classified into different experience classes.

2.4 Review Helpfulness Prediction

Many literature works (e.g. [Liu et al., 2008; Danescu-Niculescu-Mizil et al., 2009]) have suggested the helpfulness of consumer-generated reviews should be defined as Does or in what degree a review help you in making a purchase decision? If a system automatically generate helpfulness for each review, this potential consumers would assist by high quality reviews to make easy purchase decisions. The literature on evaluating the quality and helpfulness of reviews or posts on web forums is surprisingly small. Kim et al. [2006] deliver a method to automatically assess review helpfulness. They use SVR to train their system and find that the length of the review, the unigrams and the product rating are the most important features. Weimer et al.

[2007] propose an automatic algorithm to assess the quality of posts in web forums using features such as surface, lexical, syntactic, forum specific and similarity features. In [Weimer, 2007], the proposed method is examined on three datasets and finds that the SVM classification performs very well.

Liu et al. [2008] present a nonlinear regression model for the helpfulness prediction. Three groups of factors that might affect the value of helpfulness are analyzed and the model is built upon these three groups of factors. The results of applying their model show that the performance is better than the SVM regression model. Zhang and Varadarajan [2006] incorporate a diverse set of features in an attempt to build a regression model to predict the utility of online product reviews. As discussed in Chapter 1.2.1, these modeling approaches lack principles and can only learn helpfulness from a small section of available consumer-generated reviews. Our study focuses on proposing a generalized framework for analyzing the helpfulness of customer-generated reviews and thus helping consumers find the most helpful reviews more efficiently. In this generalized framework, a theoretical interpretation and a mathematical estimation technique are provided to model the helpfulness distribution of consumer-generated reviews. Furthermore, this framework unifies inference for helpfulness prediction problem and provides methods for evaluation the performance of models under this unified framework.

2.5 Probabilistic Modeling Approaches

The probabilistic (statistical) modeling approach is generally based on stochastic models [Zhou and He, 2008] which estimate the probability density of getting a certain result. It also offers principles and algorithms for estimating the parameters of each model from data [Jebara and Meila, 2006]. Under the probabilistic configuration, learning can be seen as an estimation of joint probability density functions given a set of samples. Probabilistic modeling approaches can be further categorized

into generative and discriminative [Jebara, 2003]. Generative approaches learn models of joint probability of input features and targets and then compute the posterior probability by using the Bayes rules. In contrast to the generative approach, discriminative approaches directly learn on the dependency of the learning target on the given feature.

Probabilistic modeling of text documents has been used to enhance the representation of documents. For example, probabilistic LSA (LSA) [Hofmann, 1999b,a] and Latent Dirichlet Allocation (LDA) [Blei et al., 2003] are quickly becoming the most powerful probabilistic document modeling techniques and are accepted by a variety of text processing applications [Lu et al., 2009; Karimzadehgan et al., 2008; Cai et al., 2008; Lu and Zhai, 2008; Mei et al., 2007; Andrzejewski and Zhu, 2009; Krestel et al., 2009; Bíró et al., 2009]. In probabilistic Latent Semantic Analysis (pLSA), the co-occurrence of word and document is considered as a mixture of conditionally independent multinomial distributions. In Latent Dirchlet Analysis (LDA), words are assumed to be generated independently from each other. These methodologies fall under the bag-of-word language model that assumes words are generated independently from each other. To consider the order of words an n-gram [1] based topic model [Wang et al., 2007] is proposed to discover topic phrases. Author topic model [Rosen-Zvi et al., 2004] extends LDA by incorporating authorship as an additional variable.

Many online topic models [Blei and Lafferty, 2006; AlSumait et al., 2008; Iwata et al., 2010] also have been proposed to dynamically determine topic clusters. In order to simplify the estimation, joint distribution is often required to be represented by marginal and conditional distributions. We need to make assumptions about the distribution, independence or conditional independence. In most of the above-mentioned probabilistic models, a graphical model [Buntine, 1995] is introduced to represent joint distributions using a set of dependencies specified by a graph. Graphical models use

[1] N-gram is a subsequence of n words of a given document.

a graph to denote the conditional independence between random variables. Directed Graphical models use a directed graph to represent the dependency between random variables where the notation of directionality arcs denotes the dependency of the nodes. This graphical model is also called the Bayesian Network [Heckerman et al., 1995]. When the conditional probability distribution is specified for each node, the knowledge is encoded into the model design and thus a directed acyclic graph model translates qualitative knowledge into a quantitatively representable graphical structure.

Although probabilistic modeling approaches have been successfully applied to text mining, there is no existing literature on applying probabilistic modeling approaches to the helpfulness prediction problem. The existing helpfulness prediction approaches all use the positive vote fraction as the helpfulness metric and use such a metric to train a learning machine and infer the dependency of positive vote fractions on review documents. Positive vote fractions is a poor indication of the statistics of the reader population. For example, using positive vote fraction as helpfulness metric, the fractions $\frac{1}{3}$, $\frac{9}{27}$, and $\frac{30}{90}$ are all equal, but one can hardly reason that the corresponding review documents are equally helpful. We argue that the helpfulness of a review should only depend on the review document itself and on the statistics of the reader population. Our study focuses on analyzing the helpfulness of consumer-generated reviews based on probabilistic methodology and helping users find the most helpful reviews more efficiently.

Chapter 3

Entropy Based Model

Many e-commerce web sites like Amazon.com provide platforms for users to review products and share their opinions in an effort to help consumers make the most desirable purchase decisions. However, the quality and the level of helpfulness of different product reviews are not disclosed to consumers unless they carefully analyze an immense number of lengthy reviews. Considering the large amount of available online product reviews, this is an enormous undertaking for any consumer. As a result, it is of vital importance to develop recommender systems that can evaluate online product reviews effectively to recommend the most useful items to consumers.

This chapter proposes an entropy-based model to predict the helpfulness of online product reviews, with the aim of suggesting the most suitable products and vendors to consumers. Reviews are analyzed and ranked by our scoring model and reviews that help consumers better than others will be found. In addition, we also compare our model with several machine learning algorithms. Our experimental results show that our approach is effective in ranking and classifying online product reviews.

3.1 Introduction

Recommenders are software systems that provide recommendations to assist potential buyers in choosing between diverse products and complex information. The underlying techniques (based on which recommender systems are built) range from a personalization approach, leveraging similarities between people (e.g., collaborative filtering recommenders) to an approach that exploits the knowledge base of a product domain (e.g., knowledge-based recommenders). This includes a number of different hybrid approaches. Typically, a recommender system is offered by an online store and only provides consumers with local recommendations about products at that store. This traditional type of recommender systems may not be very useful because current consumers want to take advantages of social webs to make better informed purchasing decisions by considering opinions and reviews of users from online communities and forums.

To support this trend, online review aggregation websites like Epinion.com provide platforms for consumers to exchange their opinions about products, services, and merchants. "Online product reviews provided by consumers who previously purchased products have become a major information source for consumers and marketers regarding product quality" [Hu et al., 2008].

There is some available research focused on topical categorization, sentiment classification and polarity identification of product reviews. In contrast, our work focuses on the modeling of consumer review helpfulness. Our model ranks reviews and returns an ordered list of reviews with the helpfulness estimates. Reviews provided by all members of the community are analyzed and helpful opinions are presented to consumers. We examined the performance of our model on a set of reviews collected from Amazon.com. Our experimental evaluation shows that our proposed approach outperforms or performs other machine learning methods.

The remainder of this chapter is organized as follows: Section 2 presents our

proposed approach in details. Section 3 shows our experimental evaluation including the comparison between the proposed model and other machine learning methods. Section 4 provides further discussion on the value and applications of the proposed model, and finally, Section 5 concludes the the study in this chapter.

3.2 The Proposed Approach

Our work focuses on analyzing product reviews and finding high quality and helpful reviews. In this section, we begin with a discussion on how to estimate the helpfulness of reviews and define the helpfulness function. From there, we will introduce the entropy and information gain concept before finally presenting our proposed prediction computation.

People commonly perform the following steps before deciding to buy a product: They perceive a need, seek information, compare products, consider certain user reviews, and choose a store for their purchase. Since users share reviews collaboratively, our system filters out useful information based on these reviews. We believe that the aggregation of community members' opinions by an effective model can generate good recommendations to help consumers with their purchase decision making.

3.2.1 Review Helpfulness

Consumers publish their reviews about products or services online after they have purchased and used products. Most often, consumers submit their reviews to websites like Epinion.com for potential consumers to review. Moreover, consumers can vote a review as "Helpful" or "Not Helpful" after they read the review.

Let C be the set of consumers, P be the set of products, D be the set of review documents, and V be the set of votes indicating voter opinions about the reviews (possible votes consist of "Helpful", "Not Helpful" and "Null").

- Consumer $C = \{c_1, c_2, c_3, \ldots, c_m\}$
- Product $P = \{p_1, p_2, p_3, \ldots, p_w\}$
- Review $D = \{d_1, d_2, d_3, \ldots, d_p\}$
- Vote V is a matrix listed as follows:

$$\begin{pmatrix} v_{c_1,d_1} & v_{c_1,d_2} & \ldots & v_{c_1,d_p} \\ v_{c_2,d_1} & v_{c_2,d_2} & \ldots & v_{c_2,d_p} \\ \vdots & \vdots & \vdots & \vdots \\ v_{c_m,d_1} & v_{c_m,d_2} & \ldots & v_{c_m,d_p} \end{pmatrix}$$

$$v_{c_k,d_i} = \begin{cases} Helpful & \text{if } c_k \text{ voted } d_i \text{ as Helpful,} \\ NotHelpful & \text{if } c_k \text{ voted } d_i \text{ as Not Helpful, or} \\ Null & \text{if } c_k \text{ has not voted for } d_i. \end{cases} \tag{3.2.1}$$

Definition Review helpfulness is the perception that the review $d \in D$ can be used to assist the consumers to understand the product $p \in P$. For a review d_i, its helpfulness can be calculated as the ratio of the number of consumers who have voted d_i as "Helpful" to the total number of consumers who have voted for d_i.

Let the set of all the "Helpful" votes about review d_i be denoted as h_i and the set of all "Not Helpful" votes about review d_i be denoted as $\bar{h}_i$. We define review d_i's Helpfulness as:

$$\frac{|h_i|}{|\bar{h}_i| + |h_i|} \tag{3.2.2}$$

The expected helpfulness function of an online review is a mapping, $Score : D \mapsto \mathbb{R}$. The higher score a review gets, the more useful the review is.

It is clear that given an expected helpfulness function *Score* we can sort reviews based on their scores. Furthermore, by setting a threshold θ, all reviews with score $Score > \theta$ are "Helpful" reviews and the remaining reviews are "Not Helpful".

An online review consists of words that include opinion words, words about product features, product parts and other words. The importance of each word to the helpfulness of a review can be calculated from training data that contains the vote information provided by consumers.

In the following subsection, entropy and information gain will be introduced while the reason entropy and information gain can be used to calculate the importance of words will be discussed.

3.2.2 Entropy and Information Gain

In [Pang et al., 2002], the authors reported the best result was obtained by using boolean values of unigram features. Motivated by this approach, we use the bag of words model to represent text and build our language model. Each feature is a non-stop stemmed word and the value of the feature is a boolean value of the occurrence of the word on a review.

We introduce the Shannon's information entropy concept [Shannon, 2001] to measure the amount of information in reviews. Entropy was first developed for communication and it has been used in many areas. It can be seen as the certainty of an event or the amount of information needed to represent an event. For the online review classification problem, the entropy can be extended as follows:

Let $S = \{s_1, s_2, \ldots, s_q\}$ be the set of categories in the online review space. The expected information needed to classify a review is:

$$H(S) = -\sum_{i=1}^{q} p(s_i) \log p(s_i) \tag{3.2.3}$$

The average amount of information contributed by a term t in a class s_i will be:

$$H(S|t) = -\sum_{i=1}^{q} p(s_i|t) \log p(s_i|t) \tag{3.2.4}$$

Information Gain is derived from entropy. It is originally defined as how many bits would be saved if both ends know the existence of an instance. It is often used to evaluate the relevant degree of attribute when building a decision tree [Wu et al., 2008]. In the text classification, it can be understood as the expected entropy reduction by knowing the existence of a term t. In the area of text mining and text classification, information gain is the amount of information provided by a term.

$$G(t) = H(S) - H(S|t) \tag{3.2.5}$$

Information gain is often employed as a term's goodness criterion in the field of machine learning [Yang and Pedersen, 1997]. It is used as a feature selection method in text classification and Information Retrieval deduct the dimension of documents [Anagnostopoulos et al., 2008; Lee and Lee, 2006]. In [Yang and Pedersen, 1997], information gain of term t is extended and defined as follows:

$$\begin{aligned} G(t) = -\sum_{i=1}^{q} p(s_i) \log p(s_i) + p(t) \sum_{i=1}^{q} p(s_i|t) \log p(s_i|t) \\ + p(\bar{t}) \sum_{i=1}^{q} p(s_i|\bar{t}) \log p(s_i|\bar{t}) \end{aligned} \tag{3.2.6}$$

In the above equations:

- $p(s_i)$ is the probability of documents in category s_i among all documents,
- $p(t)$ is the probability of documents which contain term t among all documents,
- $p(s_i|t)$ is the probability of documents which contain term t and that is in category s_i out of all documents containing t, and

- $p(s_i|\bar{t})$ is the probability of documents that do not contain term t and that belong to category s_i out of all documents that do not contain t.

The above formula calculates the reduction of entropy by knowing the occurrence of a specified term. It considers not only the term's occurrence, but also the term's non-occurrence. This value can indicate the term's contribution and predicting ability. If a word has higher information gain, it has more contribution to the classifying. For binary classification, information gain can be used to measure the amount of contribution of this term to a class.

Table 3.1: Information gain value example

term	Not Helpful	Helpful	Information Gain
nuvi	55	174	0.086522889
bluetooth	11	93	0.072981165
mount	13	96	0.071278275
screen	33	118	0.055793201
crash	10	2	-0.004942425
uninstal	7	0	-0.008155578
minimum	10	0	-0.011693703

In our case, only two categories,"Helpful" and "Not Helpful", will be considered. Let s_1 be "Not Helpful" and s_2 be "Helpful". In order to provide the difference of prediction ability for two categories, a change is introduced as follows: if $p(s_1|t) < p(s_2|t)$ then $Gain(t) = G(t)$, otherwise $Gain(t) = -G(t)$. By introducing this operation, the contribution of words towards "Helpful" or "Not Helpful" can be enhanced. So, the gain value calculation in our model is:

$$Gain(t) = \begin{cases} G(t) & \text{if } p(s_1|t) < p(s_2|t), \\ -G(t) & \text{otherwise.} \end{cases} \tag{3.2.7}$$

where s_1 is the category of "Not Helpful" and s_2 is the category of "Helpful".

Thus, the importance and the prediction ability of words can be calculated by equation (3.2.7). Table 3.1 shows an example of information gain values. The second and the third columns are the occurring times of the specific term in the "Helpful" and "Not Helpful" domains, respectively. We will calculate the information gain values for all the terms that have occurred in the review documents.

3.2.3 Prediction Computation

From the discussion above, the Gain value could represent the words' ability of correctly predicting if a document belongs to "Helpful" or "Not Helpful" reviews. We use the summation of the Gain values of all words in a review to indicate the review's helpfulness. In our approach, a review's content (words) will be analyzed and the Gain value will be calculated for each word (excluding stop words) of the review. As a result, to calculate the helpfulness of a review d_i, we propose the score calculation equation as follows:

$$Score(d_i) = \sum_{j=1}^{M} Gain(t_j) * f(d_i, t_j) \tag{3.2.8}$$

where $Gain(t_j)$ is the j^{th} stemmed word's Gain value; M is the total number of stemmed words in review d_i, and

$$f(d_i, t_j) = \begin{cases} 1 & \text{if term } t_j \text{ occurrs in } d_i, \text{ or} \\ 0 & \text{if term } t_j \text{ does not occurre in } d_i. \end{cases} \tag{3.2.9}$$

Equation (3.2.8) can be seen as the total helpfulness information delivered by review d_i. This equation can be used as a model to predict the helpfulness. All the score values of reviews $d_i \in R$ will be calculated. As a result, tuples of the form $< d_i, Score(d_i) >$ will be returned by this approach where d_i is an online product review and $Score(d_i)$ is the review d_i's helpfulness value. Finally, online product

reviews are ranked based on their corresponding $Score(d_i)$ values as reviews with higher *Score* values are more helpful than others. With a set **T** of training reviews of a specific product category, and a set **T'** of test reviews, the helpfulness prediction process will be as follows:

1. Find the *Gain* values for every non-stop word from **T**.
2. Calculate the *Score* value for every review of **T'** by equation (3.2.8).
3. Sort **T'** in descending order based on their *Score* values.

3.3 Experimental Evaluation

In this section we begin by introducing the evaluation method used in our experiments. We go on to describe the dataset and the experimental steps before analyzing the experimental results and evaluating the performance of our approach for classification and ranking.

3.3.1 Evaluation Method

In order to evaluate the performance of our model, precision and recall rate are used. Precision and recall are commonly used in evaluating information retrieval systems. Precision is defined as the ratio of retrieved helpful reviews to the total number of review retrieved. Recall is defined as the ratio of the number of retrieved helpful reviews to the total number of helpful reviews. "Precision and recall are thought of as some degree of correction and completeness of result" [Euzenat, 2007]. In our helpfulness discovery scenario, precision is defined as the number of helpful reviews obtained by an algorithm divided by the total number of reviews returned by an algorithm as helpful; recall is defined as the number of helpful reviews obtained by

an algorithm divided by the total number of reviews which is labeled as helpful in the dataset. The precision and recall in our case can be formulated as:

$$Precision = \frac{Reviews\ found\ and\ helpful}{Total\ reviews\ found} \tag{3.3.1}$$

$$Recall = \frac{Reviews\ found\ and\ helpful}{Total\ helpful\ reviews} \tag{3.3.2}$$

There exists trade-off between these two metrics. The increase of precision leads to the decrease of recall and the increase of recall will lead to the decrease of precision [Yuwono and Lee, 1996]. The other commonly-used performance measure is F-Measure or F-Score. F-Measure is defined as the harmonic mean of the above two measures and is calculated by

$$F = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.3.3}$$

F-Measure can evaluate the overall performance of an information retrieval approach. A higher value of F-Measure indicates a better performance of an algorithm.

We also apply Spearman's Rank Correlation Coefficient [Brazdil and Soares, 2000] between ranks to evaluate the ranking performance of our model. The list of test reviews in descending order of the percentage of positive votes is considered as a benchmark for our experiments. Rank correlation coefficient is one of the most common methods to compare two rankings on the same set of items in statistics. If the correlation between two rankings is perfect, the value is 1. If the two rankings are totally diverse from each other, the value is -1. This method will assess the relationship between predicted rankings and real rankings. The correlation of two variables X and Y can be calculated by:

$$\rho = 1 - \frac{6 \sum_i^n (x_i - y_i)^2}{n(n^2 - 1)} \tag{3.3.4}$$

where n is the number of values in each set, x_i and y_i are the benchmark rank and generated ranks of the review i in the testing set respectively.

To determine the statistical significance of the Spearman's Rank Correlation Coefficient, a t-test can be performed to test the null hypothesis of zero correlation ($\rho = 0$). The t-value is given by the following formula:

$$t = \rho\sqrt{\frac{(n-2)}{1-\rho^2}} \tag{3.3.5}$$

3.3.2 Dataset

We crawled 9955 GPS and MP3 player reviews from Amazon.com. Each of the reviews have been evaluated by at least four consumers as "Helpful" or "Not Helpful". We define that if the helpfulness of a review (percentage of helpful votes) is greater than 60 percent, the review will be marked as helpful, otherwise it is not helpful. 720 GPS reviews and 800 MP3 player reviews were randomly selected to perform the experiment.

After removing all stop words, the parsing and stemming to all the training reviews, a document term matrix is returned. Information gain is calculated for each term and is assigned a plus or minus sign based on the helpfulness. We refer these processed features as non-stop terms. Thus, the document term matrix associated with the Gain value of each stemmed word can be generated. With the Gain value, features and their corresponding importance can be discovered.

We use a basic test to evaluate the performance of our method. In the basic testing stage, 300 "Helpful" GPS reviews and 300 "Not Helpful" GPS reviews are randomly chosen to form the training dataset. Moreover, 60 "Helpful" GPS reviews and 60 "Not Helpful" GPS reviews are randomly selected to be utilized as the testing dataset. We also apply 10-fold cross-validation in evaluating the performance. In the 10-fold cross-validation, collected reviews are randomly divided into ten equal-sized folds. 10-fold cross-validation works as follows: we fit the model on 90% of

the reviews and then predict the helpfulness of the remaining 10% (the test reviews). This procedure is repeated 10 times and the performances on all 10 experiments is considered together to compute the overall performance. We apply all of the 1520 GPS and MP3 online reviews in the 10-fold cross-validation.

3.3.3 Results and Analysis

Figure 3.1 and Figure 3.2 show the distribution of review score values from the experimental result. Most of the "Helpful" reviews and "Not Helpful" reviews concentrate on the two ends of the score value space. "Helpful" online reviews will have greater scores and "Not Helpful" online reviews will have smaller scores. This distribution highly indicates that our *Score* function can model the helpfulness of online product reviews. Therefore, with the ranking of scores, most of helpful reviews can be retrieved on the top of the sorted review list.

The goal of our proposed algorithm in Section 3 is to calculate the helpfulness score for reviews. In order to categorize the sorted reviews into "Helpful" and "Not Helpful" with the scores returned by our algorithm, a Helpfulness Threshold is needed to be selected to build a classifier. Let N be the number of helpful reviews in the training set (where a review is said to be helpful if the percentage of helpful votes is greater than 60% as we defined above). Suppose we calculate the score values of all reviews in the training set and sort the set in descending order of the score values. Then, we define the Helpfulness Threshold to be the score value of the N^{th} review in the sorted training set.

Table 3.2 is the confusion matrix of the result from the basic test. It shows the performance of our model in the basic test. In this test, 41 out of 60 "Helpful" reviews are classified as "Helpful" and 56 out of 60 "Not Helpful" reviews are classified as "Not Helpful." This is the first try to evaluate our model. Only a small number of online reviews were involved to do the training and testing. In order to achieve overall

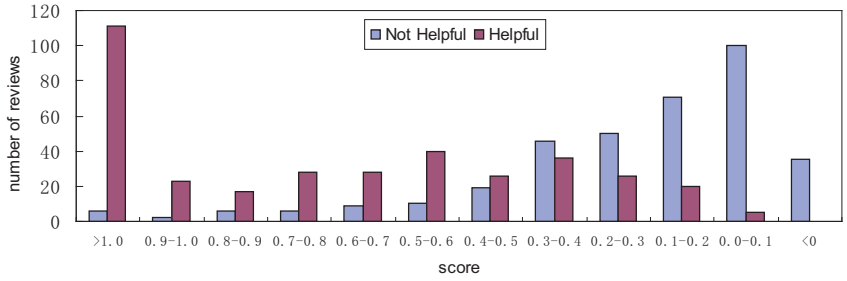


Figure 3.1: Distribution of reviews' score

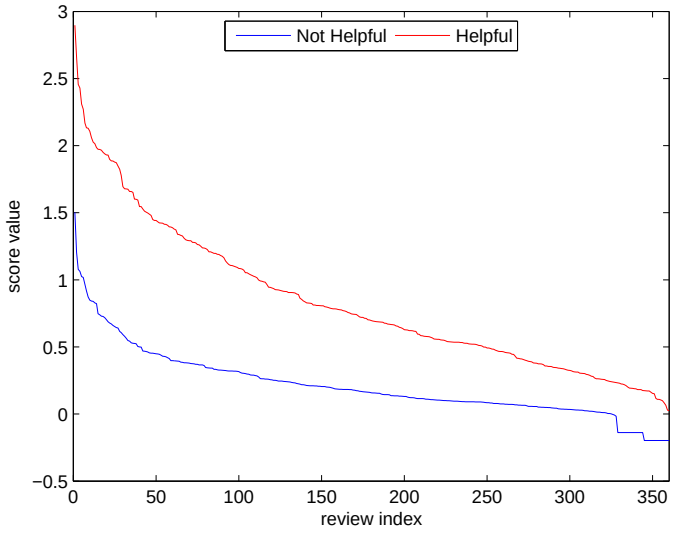


Figure 3.2: Score values of reviews

precision and recall, we perform 10-fold cross validation on a larger online review set which is described in the above subsection.

Table 3.3 presents the classification performance of our model by 10-fold cross validation. The classification precision for GPS reviews is 77.7% for "Helpful" reviews and 77% for "Not Helpful" reviews. Among the MP3 Player Reviews, the precision is 69.7% for "Not Helpful" reviews and 72% for "Helpful" reviews. It shows that the classification performance of our approach is efficient to categorize the online product reviews into "Helpful" and "Not Helpful."

Table 3.2: Confusion matrix (basic test)

(a) Training Data (300 helpful reviews, 300 not helpful reviews)

	Not Helpful	Helpful
Not Helpful	234	66
Helpful	74	226

(b) Test Data (60 helpful reviews, 60 not helpful reviews)

	Not Helpful	Helpful
Not Helpful	56	4
Helpful	19	41

Table 3.3: Precision, recall and F-measure (10-fold cross-validation)

	Precision		Recall		F-Measure	
	MP3 Player	**GPS**	MP3 Player	**GPS**	MP3 Player	**GPS**
Not Helpful	69.7%	**77%**	73.5%	**78.1%**	71.5%	**77.5%**
Helpful	72%	**77.7%**	68%	**76.7%**	69.9%	**77.2%**

Table 3.4: Performance of various classification methods and our model for GPS reviews (10-fold cross-validation)

	Precision		Recall		F-measure	
	Helpful	Not Helpful	Helpful	Not Helpful	Helpful	Not Helpful
Naive Bayes	0.747	0.768	0.778	0.736	0.762	0.752
SMO	0.796	0.749	0.728	0.814	0.761	0.78
Decision Tree	0.77	0.714	0.681	0.797	0.723	0.753
Our Model	0.777	0.77	0.767	0.781	0.772	0.775

Table 3.5: Performance of various classification methods and our model for MP3 reviews (10-fold cross-validation)

	Precision		Recall		F-measure	
	Helpful	Not Helpful	Helpful	Not Helpful	Helpful	Not Helpful
Naive Bayes	0.669	0.69	0.708	0.65	0.688	0.669
SMO	0.655	0.666	0.678	0.643	0.666	0.654
Decision Tree	0.611	0.619	0.633	0.598	0.622	0.608
Our Model	0.697	0.72	0.735	0.68	0.715	0.70

Table 3.4 is the result comparison between 10-fold cross validation performed by our model and other classification methods for GPS reviews. We compare the precision, recall, F-measure and model for Naive Bayes, Decision Tree, Sequential Minimal Optimization (SMO) and our entropy based model. The experimental result reveals that our model performs at a higher or equally compatible level as other machine learning classification methods.

For example, our approach outperforms Naive Bayes and Decision Tree. In comparison with SMO, our approach is 1% lower for the F-measure of "Not Helpful" reviews and 1% higher for the "Helpful" reviews. Table 3.5 is the result comparison between 10-fold cross validation performed by our model and other classification methods for MP3 reviews. The experiment results show that our model outperforms the Naive Bayes, Decision Tree, and SMO algorithm. The F-measure evaluation resulted by our algorithm for Helpful reviews is at least 6.5% higher than the other three algorithms that we compared with.

To evaluate a generated ranking, the Spearman's rank correlation coefficient is adopted to estimate the correlation between the predicted ranking and the real ranking in the dataset. Table 3.6 reports the ranking quality tests of Spearman rank

Table 3.6: Performance evaluation of our model ranking reviews of GPS and MP3 Players (10-fold cross-validation)

Collection	Metric	SMO Regression	Our Model
GPS	Spearman Rank Order Correlation	0.4953	0.5977
	t-value	4.9253	6.4544
MP3	Spearman Rank Order Correlation	0.3831	0.5201
	t-value	3.7253	5.4262

order correlation coefficient with t-value. It shows the correlation between the ranking manually voted by consumers and the ranking predicted by our model on the two categories, namely MP3 players and GPS. The rank correlation coefficient between the ranking generated by our model and the ranking voted by consumers is 0.5977 for GPS and 0.5201 for MP3 Player, over the 10-fold cross-validation. The t-values of 6.4544 (for GPS) and 5.4262 (for MP3) are significant at the 0.005 probabilistic level. The coefficients and t-values presented in the table suggest a significant correlation between the predicted helpfulness ranking and the original helpfulness ranking. We also compare our model with SMO Regression. It is evident from Table 3.6 that our model outperforms SMO Regression for both the GPS and MP3 reviews.

Although, in comparison with other classification and regression algorithms, our model obtains the best performance, we observe that the classification and ranking performance of our model (and also of other algorithms we compared with) for GPS reviews are better than the performance for MP3 reviews. A possible reason is that the document-term matrix of MP3 reviews is sparser and contains more noise data than the document-term matrix of GPS reviews. As a result, the information gain values of the terms in GPS reviews is greater than the information gain values of the terms in MP3 reviews. This means that the terms in GPS reviews, for which a greater information gain values are available, are more likely to appear in “Helpful”

reviews than the terms which have greater information gain values in MP3 reviews. Accordingly, the variances of the helpfulness scores of GPS reviews are larger than the variances of the helpfulness scores of MP3 reviews. The performance of classification and ranking for GPS reviews are better than that for MP3 reviews. To mitigate this problem, some external knowledge might be introduced to find the relationship between terms.

We note that the complexity of our approach is $O(|D| * W)$, where $|D|$ is the number of reviews in the training set and W is the number of non-stop words in the training set. Therefore, the system developing time for our approach is significantly lower than the other algorithms compared in our study.

3.4 Discussion and Conclusion

Traditional centralized recommender systems that are located at certain e-commerce stores, and recommend products and services local to those stores no longer serve the increasing needs of today's consumers. Consumers in our day would like to take advantage of the social web to make more informed and more effective purchasing decisions by considering the opinions of other users prior to their purchase. More and more e-commerce web sites provide facilities for users to review products and exchange opinions. User-generated product reviews have formed a rich source of information based on which satisfactory purchase decisions can be made. The entropy-based approach presented here serves as a basic step towards the goal of recommending the most useful product reviews to consumers.

The collaborative filtering approach works well to recommend ranked products by making use of user profiles. However, for our model, which is to recommend helpful reviews, the data for performing collaborative filtering experimentation is not available even if reviews are considered as a "product." The profiles of users who have voted for reviews are not available to us. The only available data are the

review documents themselves and the number of "Helpful" and "Not Helpful" votes for each review. Our model is based on the examination of the terms in a review, which reflect more "insight" of the opinions of users on the review. Whereas the collaborative filtering approach is based on the similarity and dissimilarity between user profiles. This difference allows our model to avoid some well-known disadvantages of the collaborative filtering approach, like the cold start problem and the early rater problem.

Today's online customers are also known to be more impatient and demanding than ever before. On the one hand, they would like to make the best possible purchase decisions based on as much information available. On the other hand, they want to complete their purchases in as little time as possible. In other words, customers want to receive satisfactory products and services, but they are not willing to spend much time and effort for their purchases. Therefore, we believe that our proposed model, aiming to automate the process of analyzing and ranking online product reviews to recommend the most helpful ones to consumers, is desirable.

We note that our proposed model can be tuned to process not only product reviews but also other sources of product and service related information such as user ratings, opinions and comments that can be obtained from different online user clubs, communities or forums across the Internet. It should be clear from the description of our model that it can serve as a core component in a complete and intelligent recommender system. The output of our model, i.e. the most helpful user reviews of products or services, can be used to feed as input to the second component of the system, which will then recommend the most suitable products (or services) and vendors to consumers based on the useful reviews. In fact, such a recommender system can be configured to make recommendations of either useful reviews (or similar information), or products/services and vendors, or both, depending on the user preferences.

From an online store's perspective, a recommender system can be developed based on our proposed model. This system would make use of user reviews and

similar sources of information that are available within the store's boundaries such as the store's member clubs. From a wider and across-organization perspective, different related businesses or organizations can join together to build more versatile and useful recommender systems by using our proposed approach, by accessing much larger sources of product reviews and similar information from multiple organizations as well as many available user communities and forums across the Internet.

We have proposed an entropy-based approach for modeling the helpfulness of online product reviews. The empirical results show that our model can effectively classify and rank online product reviews based on the "helpfulness score" generated by the model. Reviews with high "helpfulness scores" will be found and recommended to consumers. This review recommending system can be used to help consumers reach helpful reviews more easily and a recommender system that incorporates the function of recommending useful reviews would be more useful and attract more potential buyers.

We compared our model with other state-of-the art algorithms for classifying and ranking, and found that our model is comparable with those algorithms in GPS and MP3 reviews. In comparison with other helpfulness assessment methods, our model is simplified, easier to understand and simpler to implement. The time complexity of our model is $O(|D| * W)$, where $|D|$ is the number of reviews in the training set and W is the number of non-stop words in the training set. Therefore, the proposed model can classify and rank reviews significantly faster than other algorithms.

One limitation of this approach is the extendability of the algorithm. We mainly focus on the feature set of non-stop terms when modeling the helpfulness for online reviews. This can cause the model to not easily be extended so that a generalized framework is expected to model the helpfulness of review documents.

The other limitation of this approach is the compatibility of the algorithm. It is not reliable to make use of the positive vote fraction as the benchmark for a training purpose when a review is voted by very few voters. Not only our model, but also

other existing helpfulness assessment algorithms are utilizing review documents with a large number of voters as the training data. In reality, most of the reviews are only voted by a small number of voters which limits the compatibility of our model.

Our entropy-based approach is relatively simple, efficient and suitable for applications having fully-voted reviews available and requiring the recommendation engine to be rapidly developed. In the next chapter, we will present a more robust formulation of review helpfulness with precise mathematical principals suitable for learning from both sparsely-voted reviews and fully-voted reviews to overcome the compatibility and extensibility of existing approaches. We treat user opinions (reader votes) in a probabilistic manner, naturally taking into account the uncertainty arising from "words of few mouths". We also develop a probabilistic approach and demonstrate that it significantly outperforms prior arts in this setting.

Chapter 4

Declarative Probabilistic Model

This chapter presents a rigorous probabilistic framework for inferring the "helpfulness" of consumer-generated reviews which can build a "helpfulness" model from consumer votes and consumer-generated reviews. Under this framework, the "helpfulness" of a review document is given a precise mathematical meaning. Also, we introduce a new quantitative helpfulness metric as the benchmark for the "helpfulness" of consumer-generated review documents. Our experimental results show that it performs superior to the existing helpfulness metric. This framework, by formulating helpfulness inference as an optimization problem, provides a natural methodology for evaluating and comparing helpfulness inference algorithms. Furthermore, the framework is not limited to a low number of votes on a review document, a situation that can hardly be handled by existing models. In addition, we develop, under this framework, a probabilistic graphical model and an Expectation Maximization algorithm for helpfulness inference, to demonstrate the versatility by specifying dependencies between framework components. Our algorithm is compared experimentally to other existing helpfulness discovering algorithms and the experimental results show that our framework can effectively model the helpfulness of consumer-generated reviews better than other existing approaches, and therefore, indicate the capability of our

framework to recommend helpful consumer-generated reviews to the potential consumers. Although this framework primarily considers the problem of review helpfulness prediction, the presented probabilistic methodology is in general applicable for developing other recommender systems based on voter opinions.

4.1 Introduction

One of the characteristic features of an electronic shopping environment is that there exists a vast amount of information available when making purchase decisions [Parra and Ruiz, 2009]. The consumer-generated review is more and more prevalent in most of the electronic commerce web sites, like Amazon.com, that provide not only products and services, but also online reviews or consumer-generated reviews, composed by previous purchasers. It is well known that consumers need to find out the trustable experiences from all kinds of consumer-generated reviews [Duan et al., 2008]. By taking the advantages of the increasing availability of rich information, consumers enjoy a greater number of experiences than ever before. Still, the quality of the online reviews varies significantly and such a large amount of information may overburden any consumer and due to the unavailability of the helpfulness of consumer-generated reviews. Currently, many opinion sharing platforms provide a function to let readers vote for helpful or not about a review or opinion post. Potential consumers can make use of this information to estimate the helpfulness of the review and decide to read it or not. As the vote collection process takes time and newly posted reviews always get fewer votes, a system that can easily discover helpful online reviews is much needed to reduce the time of acquiring helpful reviews.

To automatically recommend helpful consumer-generated reviews to potential buyers, a number of helpfulness assessment approaches have been developed to model the helpfulness value of consumer-generated reviews [Kim et al., 2006; Liu et al., 2008; Zhang and Varadarajan, 2006]. These helpfulness assessment approaches utilize the

positive vote fraction as the helpfulness benchmark and focus on applying different machine learning tools to learn a helpfulness function which is a mapping $Score : \mathcal{D} \mapsto \mathbb{R}$, where $\mathcal{D}$ is the space of all review documents and $\mathbb{R}$ represents the set of real numbers. The major limitation of the conventional approaches is the utilization of the positive vote fraction based benchmarking methodologies. Particularly when only a limited number of votes are available, the positive vote fraction is unable to significantly represent the true helpfulness of a review document.

Aware of such a limitation, most existing approaches solely attempt to analyze the consumer-generated reviews for which a relatively large number of votes are available. However, even when a large number of voters' opinions are available for each consumer-generated review, the same helpfulness value of positive vote fraction can come from different sizes of voter population and this fraction fails to capture the confidence of the helpfulness estimates. Furthermore, as we discussed in Chapter 1, the "words of few mouths" is a widely existing phenomenon in the context of recommender system development based on user opinions, namely, most of online consumer reviews receive very few votes (see Figure 1.2 for the histogram of the number of votes on a set of review documents taken from Amazon.com). This phenomenon presents additional challenges for developing machine-learning algorithms in recommender systems, since the very few users' opinions, if treated improperly, are either un-utilized, leading to lack of resources for learning, or becoming an additional source of "noise" in the training of the algorithms. The applicability of the existing approaches therefore is significantly limited to a small part of available information.

In our approach, instead of considering user opinions on "unpopular items" unreliable and throwing them away, we treat user opinions in a probabilistic manner, naturally taking into account the uncertainty arising from "words of few mouths". Specific to the review helpfulness prediction problem, we develop a probabilistic approach and demonstrate that it significantly outperforms prior arts in this setting. In this chapter, we present a more robust formulation of review helpfulness with

precise mathematical meaning, and a more principled probabilistic helpfulness assessment framework for inferring the helpfulness distribution of consumer-generated review using the distribution of helpfulness conditioned on the feature of the document to characterize the dependency of the helpfulness and the review. Furthermore, we introduce a new quantitative helpfulness metric to capture the uncertainty of the helpfulness estimates and a new probabilistic helpfulness metric based on which our model can rank consumer-generated reviews by the obtained helpfulness distribution.

Moreover, there are currently no standard evaluation metrics to judge the performance of review helpfulness assessment algorithms. In this chapter, we also discuss the evaluation aspects of the review helpfulness modeling framework and propose a uniformed evaluation measurement for assessing the success of helpfulness prediction models. Experiments on review data sets of three product categories from Amazon.com suggest that our algorithm in fact outperforms the previously reported prediction algorithms and show that the algorithm can effectively predict the helpfulness distribution of consumer-generated review and recommended the most helpful contents to potential consumers.

The remainder of this chapter is organized as follows. Section 2 introduces our framework to estimate the helpfulness distribution of a review and how to evaluate the performance of a helpfulness assessing model. Section 3 delivers a graphical model based approach under our framework. Section 4 presents experimental evaluation of our framework. Finally, Section 5 will discuss the advantages of our proposed framework and conclude this chapter.

Table 4.1: Table of notations

$\mathcal{D}$	the space of all review documents
$\mathbb{R}$	the set of real numbers
I	the finite indexing set of available reviews
d_I	the set of all available reviews
d_i	a review document
J	the finite indexing set of available votes
v_J	the set of all available votes
v_j	a voter opinion
$J(i)$	the set of all votes concerning review d_i
$J^+(i)$	the positive (i.e., `HELPFUL`) votes on review i
$J^-(i)$	the negative (i.e., `UNHELPFUL`) votes on review i
α_i	positive vote fraction
$\mathbf{V}$	random variable of votes
$\mathbf{D}$	random variable of review documents
v	value of $\mathbf{V}$
d	value of $\mathbf{D}$
$\mathcal{D}$	the space of all consumer-generated reviews
$\mathcal{W}$	the population of readers
$\mathcal{R}$	a family of functions mapping $\mathcal{D}$ to $\{0, 1\}$
$\mathcal{P}$	a probability measure on $\mathcal{R}$
$r \in \mathcal{R}$	a rule of voting
$\alpha(d)$	the probability that document d will be voted positively

4.2 Probabilistic Framework of Review Helpfulness

This section presents a helpfulness discovering model by utilizing the graphical model and an expectation maximization (EM) algorithm under our proposed framework. The graphical model is used to represent the dependencies between components in the helpfulness discovering model and EM algorithm is applied to estimate the parameters.

4.2.1 Problem Statement and Conventional Approaches

The problem of predicting the helpfulness of an online review may be simply phrased as the following problem: Given a relatively large set of online reviews generated by the users and a collection of online readers' votes (in terms of `HELPFUL` or `UNHELPFUL`) on the reviews, determine how "helpful" a new review would be to the users when no readers' votes are yet available. For readers' convenience, Table 4.1 lists complete notations used in Chapter 4 and 5.

Formally, we use $d_I := \{d_i : i \in I\}$ to denote the set of all available reviews, where set I is a finite indexing set and each d_i, $i \in I$ is a review document. Similarly, we use $v_J := \{v_j : j \in J\}$ to denote the set of all available votes, where set J is another finite indexing set and each $v_j, j \in J$, is $\{0, 1\}$-valued variable, or a vote, with 1 corresponding to `HELPFUL` and 0 corresponding to `UNHELPFUL`. The association between votes and reviews effectively induces a partition of index set J into disjoint subsets $\{J(i), i \in I\}$, where for each i, $J(i)$ indexes the set of all votes concerning review d_i. In particular, each set $J(i)$ naturally splits into two disjoint subsets $J^+(i)$ and $J^-(i)$, indexing respectively the positive (i.e., `HELPFUL`) votes on review i and the negative (i.e., `UNHELPFUL`) votes on review i.

The helpfulness prediction problem of the interest of this paper can then be

rephrased as determining how helpful an arbitrary review d, not necessarily in d_I, would be, given d_I, v_J and the partition $\{J(i) : i \in I\}$.

To arrive at a mathematical formulation of the problem, what remains to characterize is the meaning of "helpfulness". Conventional approaches (see, for example, [Kim et al., 2006] [Weimer, 2007] [Liu et al., 2008], ... etc) characterize the helpfulness of review i as the fraction of votes indexed by $J(i)$ that are equal to 1. This measure, which we call *positive vote fraction* of review i and denote it by α_i, may be formally defined as follows.

$$\alpha_i = \frac{|J^+(i)|}{|J(i)|}, \tag{4.2.1}$$

where $|\cdot|$ denotes the cardinality of a set.

Built on this measure of helpfulness, conventional approaches, including for example, SVR and ANN, start with extracting the positive vote fraction α_i for each review in d_I and attempt to infer the dependency of positive vote fraction α on a generic document d. These approaches are deterministic in nature, since they all assume a *functional* dependency of α on d. The methodology of these approaches boils down to first prescribing a family of candidate functions describing this dependency and then, via training using data $(d_I, v_J, \{J(i) : i \in I\})$, selecting one of the functions that best fit the data.

Despite promising results reported for several cases, these approaches are fundamentally limited due to the improper characterization of "helpfulness". To see this, we first note that the helpfulness of a review should only depend on the review document itself and on the *statistics* of the readers (characterized by an unknown distribution over the family of all possible ways that a reader may use to decide on his vote). Although the positive vote fraction may approximate, to a certain degree, the helpfulness of a review document and this approximation may be justified under asymptotic assumptions, in real-life scenarios where the number of votes on a review can be very low (see Figure 1.2 for a histogram of number of votes on reviews

from Amazon.com), characterizing helpfulness as a positive vote fraction is highly unreliable.

In addition, we remark that positive vote fractions are *not* sufficient statistics of voting data. For example, a positive vote fraction of $\frac{3}{5}$ resulting from 3 positive votes out of a total of 5 votes are clearly less meaningful than the same positive vote fraction resulting from 30 positive votes out of a total of 50 votes, and it is hardly justifiable that in this case, the two documents have the same "helpfulness" value.

At this end, we see the need of a more robust formulation of review helpfulness and a more principled approach to helpfulness prediction. Prediction algorithms deviating from the conventional positive vote fraction based methodology are clearly desirable.

4.2.2 Probabilistic Framework

In this section, we introduce a probabilistic framework for the inference of review helpfulness. Throughout this chapter, a random variable will be denoted by a bold-font capitalized letter, for example $\mathbf{V}$, $\mathbf{D}$; and any value the random variable may assume will be denoted by its corresponding lower-cased letter, for example v, d. The distribution (either probability mass function or probability density function) of a random variable, say, $\mathbf{X}$, will be denoted by $p_{\mathbf{X}}$. Following the above mentioned notational convention, when $p_{\mathbf{X}}$ is treated as a function, we also write as $p_{\mathbf{X}}(x)$ when necessary, namely using the lower-cased letter "x" as the argument of the function.

We will often encounter a collection of random variables, say, $\{\mathbf{X}_k : k \in K\}$ for some index set K. In this case, we may write the set of random variables collectively as $\mathbf{X}_K$, and its corresponding (vector) value as x_K.

Probabilistic formulation of helpfulness

We will use $\mathcal{D}$ to denote the space of all consumer-generated reviews and use $\mathcal{W}$ to denote the population of readers who may vote on the documents in $\mathcal{D}$. We argue that the "helpfulness" of any document $d \in \mathcal{D}$ is not only a property of document d itself, but it, in fact, also depends on $\mathcal{W}$. Specifically, a given document d may be more helpful with respect to one population $\mathcal{W}$ than it is with respect to another population $\mathcal{W}'$.

To rigorously define "helpfulness", we characterize $\mathcal{W}$ as a pair $(\mathcal{R}, \mathcal{P})$, where $\mathcal{R}$ is a family of functions mapping $\mathcal{D}$ to $\{0, 1\}$ and $\mathcal{P}$ is a probability measure on $\mathcal{R}$. Here, each function $r \in \mathcal{R}$ specifies a rule of voting, whereby $r(d) = 1$ indicates that the document d will be voted positively (ie, as `HELPFUL`) under rule r. In this setting, we define the helpfulness of a document d, denoted by $\alpha(d)$, as the probability that document d will be voted positively by a random reader in $\mathcal{W}$, namely,

$$\alpha(d) := \Pr[\mathbf{R}(d) = 1],$$

where $\mathbf{R}$ is drawn at random from probability space $(\mathcal{R}, \mathcal{P})$. Equivalently, the helpfulness of document d can be characterized by the conditional probability distribution $p_{\mathbf{V}|\mathbf{D}}(v|d)$, where $\mathbf{D}$ is a random document from $\mathcal{D}$ and $\mathbf{V}$ is the $\{0, 1\}$-valued opinion of a random voter from $(\mathcal{R}, \mathcal{P})$. We are particularly concerned with the probability measure on $\mathcal{D}$ from which the random document $\mathbf{D}$ is chosen since $\mathbf{D}$ is always conditioned upon throughout the chapter. It is then clear that $p_{\mathbf{V}|\mathbf{D}}(v = 1|d) = \alpha(d)$ and $p_{\mathbf{V}|\mathbf{D}}(v = 0|d) = 1 - \alpha(d)$.

The helpfulness inference problem

Under this formulation of helpfulness, the objective of inferring the helpfulness is to learn $\alpha(d)$, or $p_{\mathbf{V}|\mathbf{D}}(v|d)$ for every $d \in \mathcal{D}$, based on a sample of documents drawn from $\mathcal{D}$ and a collection of votes on these documents.

Let $\mathbf{D}_I$ be a random set of documents drawn from $\mathcal{D}$, where each element in I is the index of a document in $\mathbf{D}_I$. For each $i \in I$, let $J(i) \subset \mathcal{W}$ denote the set of readers who have voted on document i. Denote by $J := \cup_{i \in I} J(i)$ the set of all voters that have voted on at least one document in I.

In the helpfulness inference problem, we wish to determine $p_{\mathbf{V}|\mathbf{D}}(v|d)$ when $\mathbf{D}_I$ is given as some d_I and $\mathbf{V}_J$ is given as some v_J.

Probabilistic formulation of helpfulness inference

We now present a probabilistic framework of this problem. Essential to this framework is the creation of a model that describes the relationship between documents, their helpfulness and the voter opinion. Here, a *model* is a sensible family $\Theta_{\mathbf{V}|\mathbf{D}}$ of conditional distributions of $\mathbf{V}$ given $\mathbf{D}$. The objective of helpfulness inference is to select a distribution $p_{\mathbf{V}|\mathbf{D}}$ from the family $\Theta_{\mathbf{V}|\mathbf{D}}$ such that $p_{\mathbf{V}|\mathbf{D}}(v_J|d_I)$ is maximized. That is, the problem translates to determine:

$$\begin{aligned} p^*_{\mathbf{V}|\mathbf{D}} &= \operatorname*{argmax}_{p_{\mathbf{V}|\mathbf{D}} \in \Theta_{\mathbf{V}|\mathbf{D}}} \log p_{\mathbf{V}|\mathbf{D}}(v_J|d_I) \\ &= \operatorname*{argmax}_{p_{\mathbf{V}|\mathbf{D}} \in \Theta_{\mathbf{V}|\mathbf{D}}} \log \prod_{i \in I} p_{\mathbf{V}|\mathbf{D}}(v_{J(i)}|d_i), \end{aligned} \tag{4.2.2}$$

where the latter equality is due to the fact that given document i, voters opinion on the document is independent on their opinions on any other documents.

Furthermore, we assume voters in each $J(i)$ are chosen independently, giving rise to:

$$p_{\mathbf{V}|\mathbf{D}}(v_{J(i)}|d_i) = \prod_{j \in J(i)} p_{\mathbf{V}|\mathbf{D}}(v_j|d_i) \tag{4.2.3}$$

Applying our definition of helpfulness, when a document D is drawn from $\mathcal{D}$ (under any probability measure) and a random voter V is chosen from $\mathcal{W}$ under $\mathcal{P}$, it

is easy to see that random variables D, H and V form Markov chain [1] $D - H - V$, where

$$H := \alpha(D). \tag{4.2.4}$$

Specifically,

$$p_{\mathbf{V}|\mathbf{D},\mathbf{H}}(v|d,h) = p_{\mathbf{V}|\mathbf{H}}(v|h) = h^{[v=1]}(1-h)^{[v=0]}, \tag{4.2.5}$$

where $[P]$ (known as Iverson's convention) for any Boolean proposition P evaluates to 1 if P is true, and to 0 otherwise.

Putting together Eq.(4.2.2), Eq.(4.2.3) and Eq.(4.2.5), we have:

$$\begin{aligned} p^*_{\mathbf{V}|\mathbf{D}} &= \underset{p_{\mathbf{V}|\mathbf{D}} \in \Theta_{\mathbf{V}|\mathbf{D}}}{\operatorname{argmax}} \log \prod_{i \in I} \prod_{j \in J(i)} \int p_{\mathbf{V},\mathbf{H}|\mathbf{D}}(v_j, h_i|d_i) dh_i \\ &= \underset{p_{\mathbf{V}|\mathbf{D}} \in \Theta_{\mathbf{V}|\mathbf{D}}}{\operatorname{argmax}} \left[\log \prod_{i \in I} \prod_{j \in J(i)} \int h_i^{[v_j=1]} (1-h_i)^{[v_j=0]} p_{\mathbf{H}|\mathbf{D}}(h_i|d_i) dh_i \right], \end{aligned} \tag{4.2.6}$$

where the final equality follows from:

$$\begin{aligned} p_{\mathbf{V}|\mathbf{D}}(v|d) &= \int p_{\mathbf{V},\mathbf{H}|\mathbf{D}}(v,h|d) dh \\ &= \int p_{\mathbf{V}|\mathbf{H}}(v|h) p_{\mathbf{H}|\mathbf{D}}(h|d) dh \\ &= \int h^{[v=1]} (1-h)^{[v=0]} p_{\mathbf{H}|\mathbf{D}}(h|d) dh. \end{aligned} \tag{4.2.7}$$

We note Eq.(4.2.7) also implies that determining $p^*_{\mathbf{V}|\mathbf{D}}(v|d)$ boils down to determine $p^*_{\mathbf{H}|\mathbf{D}}(h|d)$ from a family $\Theta_{\mathbf{H}|\mathbf{D}}$ of conditional distribution of $\mathbf{H}$ on $\mathbf{D}$ which are sensible and may induce the family $\Theta_{\mathbf{V}|\mathbf{D}}$ under Eq.(4.2.7). Then we reformulate the helpfulness inference problem as finding:

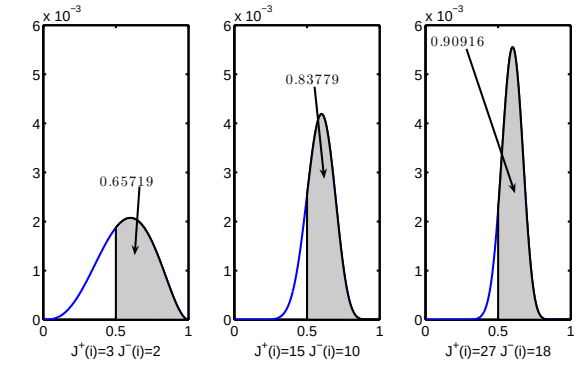
$$p^*_{\mathbf{V}|\mathbf{D}} = \underset{p_{\mathbf{H}|\mathbf{D}} \in \Theta_{\mathbf{H}|\mathbf{D}}}{\operatorname{argmax}} \left[\log \prod_{i \in I} \prod_{j \in J(i)} \int h_i^{[v_j=1]} (1-h_i)^{[v_j=0]} p_{\mathbf{H}|\mathbf{D}}(h_i|d_i) dh_i \right]. \tag{4.2.8}$$

[1] Random variables X, Y and Z are said to form Markov chain X-Y-Z if X and Z are independent conditioned on Y

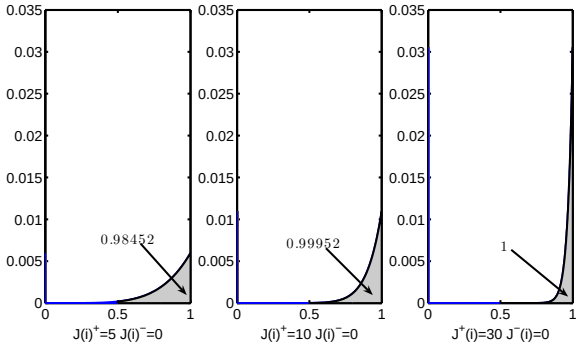
for some properly defined family $\Theta_{\mathbf{H}|\mathbf{D}}$.

It is worth noting that H depends on D functionally (as given in (4.2.4)). The nature of the problem is that neither $\mathcal{W}$ nor $\mathcal{P}$ is known. This lack of knowledge necessarily implies that an appropriate dependency model of H on D is to assume that H depends on D probabilistically. This need is further amplified when a numerous amount of votes are not available for some documents.

Helpfulness Ranking



(a) $h_i = \frac{3}{5}$



(b) $h_i = 1$

Figure 4.1: The probability density function of the review helpfulness. $J(i)^+$ and $J(i)^-$ denote the number of positive and negative votes for each review in the data set. (a) Positive vote fraction is $\frac{3}{5}$. (b) Positive vote fraction is 1.

Once helpfulness distribution $p_{\mathbf{H}|\mathbf{D}}(h_i|d_i)$ in the above mentioned framework is determined, a question which naturally arises is how to rank document according to $p_{\mathbf{H}|\mathbf{D}}(h_i|d_i)$. We define the probability of a review being helpful is greater than 0.5 as a ranking metric:

$$r_i := p_{\mathbf{H}|\mathbf{D}}(h_i > 0.5|d_i) = \int_{0.5}^{1} p_{\mathbf{H}|\mathbf{D}}(h_i|d_i)dh_i. \tag{4.2.9}$$

We label this metric as *helpfulness bias*, which describes the helpfulness distribution bias towards 0 or 1. Intuitively, this conditional probability, r_i, can be chosen as a quantitative metric to measure the helpfulness of consumer-generated review.

Suppose the "Oracle" is accessible to the learning machine, the learning machine should take $p_{\mathbf{H}|\mathbf{V}}(h_i|v_{J(i)})$ as the "best guess" for the helpfulness distribution; yet, an assumption we may make here is that without access to the "Oracle", $p_{\mathbf{H}}(h_i)$ is uniformly distributed. The $p_{\mathbf{H}|\mathbf{V}}(h_i|v_{J(i)})$ is the estimated density of h_i under learning with the "Oracle" and it is clear that:

$$\begin{aligned} p_{\mathbf{H}|\mathbf{V}}(h_i|v_{J(i)}) &= \frac{p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i)}{p_{\mathbf{V}}(v_{J(i)})} \\ &= \frac{p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i)}{\int_{h_i} p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i)dh_i}. \end{aligned} \tag{4.2.10}$$

Figure 4.1 describes the conditional probability density function $p_{\mathbf{H}|\mathbf{V}}(h_i|v_{J(i)})$ given by six different $v_{J(i)}$ examples. It can be observed that with the same helpfulness value, the degree of certainty (the width of the distribution) is completely different. When a small number of voters are given, the distribution of helpfulness is highly uncertain. When large sufficient voters voted on a document, we expect the helpfulness distribution is narrow and highly concentrated on the value of positive vote fraction.

The helpfulness bias of the "Oracle", with the prior knowledge of voter opinion, can be formulated as the integral of the posterior density over [0.5, 1], namely,

$$p_{\mathbf{H}|\mathbf{D}}(h_i > 0.5|d_i) = \int_{0.5}^{1} \prod_{j \in J(i)} h_i^{v_j} (1 - h_i)^{1 - v_j} dh_i. \tag{4.2.11}$$

The helpfulness bias of the "Oracle" equals the area under the probability curve bounded by 0.5 and 1 (as indicated by the shaded area in Figure 4.1 which shows the helpfulness distribution in different circumstances). Figure 4.1(a) demonstrates the helpfulness values for consumer-generated review that hold the same positive fraction of $\frac{3}{5}$. It can be observed that the helpfulness distributions of consumer-generated review vary significantly although the positive vote fractions are same. Figure 4.1(b) demonstrates the helpfulness values for consumer-generated review that hold the same positive fraction of 1. Conventional helpfulness formulation will fail to capture the difference between the helpfulness of those consumer-generated reviews and declare that they are the same helpful under this circumstance. In considering the problem of limitation of the positive vote fraction based benchmarking methodology, we advocate the helpfulness bias of "Oracle" to serve as the learning target for machine learning algorithms and regard it as a benchmark to compare against. With the proposed helpfulness bias as the ranking metric, the probability of the consumer-generated review being helpful is identified and is ordered by this probability.

Feature as sufficient statistics for helpfulness inference

Since this function characterizing the dependency of helpfulness on a consumer-generated review d_i is difficult to be determined in a learning framework, instead of considering helpfulness as depending on the review functionally, we may consider helpfulness as depending on a set of extractable features of the review *probabilistically*. Let $\mathcal{F}$ denote the set of all features relevant to voter opinion. That is, there is a function β mapping $\mathcal{D}$ onto $\mathcal{F}$ such that for every $d \in \mathcal{D}$ and every $v \in \mathcal{W}$, $v(d) = u_v(\beta(d))$ for some function u_v on $\mathcal{F}$. It then follows that for a randomly drawn document $\mathbf{D}$ from $\mathcal{D}$ (under any probability measure), random variables $\mathbf{D}$, $\mathbf{F}$

and $\mathbf{H}$ form a Markov chain, where $\mathbf{F} := \beta(\mathbf{D})$ namely, given the feature $\beta(\mathbf{D})$ of $\mathbf{D}$, the helpfulness of $\mathbf{D}$ is independent of the document $\mathbf{D}$ itself.

We can regard features as a sufficient statistic of the document, i.e.

$$p_{\mathbf{H}|\mathbf{D}}(h_i|d_i) = p_{\mathbf{H}|\mathbf{F},\mathbf{D}}(h_i|f_i, d_i) = p_{\mathbf{H}|\mathbf{F}}(h_i|f_i).$$

The objective of determining the helpfulness of the consumer-generated reviews then can be formulated as finding:

$$p^*_{\mathbf{V}|\mathbf{F}} = \underset{p_{\mathbf{H}|\mathbf{F}} \in \Theta_{\mathbf{H}|\mathbf{F}}}{\operatorname{argmax}} \log\left[\prod_{i \in I} \prod_{j \in J(i)} \int {h_i}^{[v_j=1]}(1-h_i)^{[v_j=0]} p_{\mathbf{H}|\mathbf{F}}(h_i|f_i)dh_i\right]. \quad (4.2.12)$$

In other words, we seek to determine $p_{\mathbf{H}|\mathbf{F}}$ from properly defined family of $\Theta_{\mathbf{H}|\mathbf{F}}$ which maximize the logarithm of $p_{\mathbf{V}|\mathbf{F}}(v_J|f_I)$.

Evaluation of Model and Estimated Parameters

Under our proposed helpfulness framework, we can design different models to find the probability distribution of voters' opinions given a feature set. It is necessary to introduce a criterion to evaluate the goodness-of-fit for these algorithms/models. As defined in Eq.(4.2.2), the logarithm of the likelihood of observing all votes collectively given the features of all documents can be seen as how the model fit the data. Therefore, the objective of building a helpfulness model in our framework is to choose parameter values that achieve the best representation of the observed data. In other words, the criterion for choosing parameters from candidate distribution parameters is to find the one that best fits the training set.

For the "Oracle" learning machine, which has the prior knowledge of voters' opinions, $p_{\mathbf{H}|\mathbf{V}}(h_i|v_{J(i)})$ is the best possible estimate for $p_{\mathbf{H}|\mathbf{F}}(h_i|f_i)$. Such, the goodness-of-fit for the "Oracle" learning machine can be formulated as:

$$p_{\mathbf{V}|\mathbf{F}}(v_J|f_I) = \prod_{i \in I} p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i = \alpha^*_i)$$

$$= \prod_{i \in I} \prod_{j \in J(i)} \alpha_i^{*[v_j=1]} (1 - \alpha_i^*)^{[v_j=0]}, \tag{4.2.13}$$

Accordingly, the difference between the logarithm of $p_{\mathbf{H}|\mathbf{V}}(h_i|v_{J(i)})$ of our algorithm and the Oracle estimation indicates the fitness of the algorithm for consumer-generated review documents.

The log-likelihood can also assist us to measure the goodness-of-fit for different models. The log-likelihood function is the logarithm of the model formulation. In our case, the fitness of a model/algorithm can be formulated as $\log p_{\mathbf{V}|\mathbf{F}}(v_J|f_I)$, where $p_{\mathbf{V}|\mathbf{F}}(v_J|f_I)$ is the marginal probability distribution learned by an algorithm.

4.3 Helpfulness Distribution Discovering Model

4.3.1 Graphical Model

"Graphical models are a marriage between probability theory and graph theory. They provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering – uncertainty and complexity – and in particular they are playing an increasingly important role in the design and analysis of machine learning algorithms. Fundamental to the idea of a graphical model is the notion of modularity – a complex system is built by combining simpler parts" [Jordan, 1999]. A graphical model represents the random variables by nodes and the conditional independencies between random variables by edges.

Bayesian Network [Heckerman et al., 1995] is a probabilistic graphical model and is represented as directed graph while the notation of directionality arcs denotes the dependency of nodes. In order to state the conditions for the parameters of the model, the conditional probability distribution should be specified for each node. Thus, the knowledge is encoded into the model design and this directed acyclic graph model translates qualitative knowledge into a quantitatively representable graphical

structure. Also, the model built with this approach has physical interpretations. It can reduce the dimensionality by simplifying the complexity of the joint distribution through the independency specified in the graph. The learning/inference processes and results can be interpreted in a probabilistic sense. As the ability of modeling stochasticity and representing dependencies between variables, we make use of the Bayesian Network to construct the helpfulness discovery model.

4.3.2 Generic EM Algorithm

This subsection describes the expectation maximization (EM) algorithm [Dempster et al., 1977b], that maximizes the incomplete likelihood function. Let $\mathbf{X} := (\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_N})$ be the set of observed random variables, $\mathbf{Z} := (\mathbf{Z_1}, \mathbf{Z_2}, \ldots, \mathbf{Z_N})$ be the set of missing data or so called hidden random variables, and $\theta := (\theta_\mathbf{1}, \theta_\mathbf{2}, \ldots, \theta_\mathbf{M})$ be the set of unknown parameters. Suppose that we are given the parametric form of $p(x, z|\theta)$, and we want to estimate θ based on the observation x. That is, we want to find $\hat{\theta}$ which maximizes the incomplete log-likelihood of:

$$l(x;\theta) = \log p(x|\theta) = \log \sum_z p(x, z|\theta).$$

where $p(x|\theta)$ is the probability of x given parameter θ and $p(x, z|\theta)$ is the joint distribution of x and z given θ.

In fact, with the EM algorithm, we can obtain the estimates of both θ and z. Let $q(z)$ be an arbitrary distribution of z, that does not carry any physical meaning but only serves to "unfold another dimension" for function $l(\theta; x)$. We then have

$$\begin{aligned} l(\theta;x) = \log \sum_z q(z) \frac{p(x,z)}{q(z)} &\geq \sum_z q(z) \log \frac{p(x,z)}{q(z)} \\ &:= \mathcal{L}(q,\theta). \end{aligned}$$

The EM algorithm is essentially a coordinate ascent algorithm on $\mathcal{L}$. That is, in

the E-step, we fix θ and find q to maximize $\mathcal{L}$; and in the M-step, we fix q and find θ to maximize $\mathcal{L}$. The following two steps formulate the process of the EM algorithm.

E-Step:

$$q^{t+1} := \operatorname*{argmax}_{q} \mathcal{L}(q, \theta^t) = p(z|x, \theta^t)$$

M-Step:

$$\theta^{t+1} = \operatorname*{argmax}_{q} \mathcal{L}(q^{t+1}, \theta) = \operatorname*{argmax}_{\theta} \sum_{z} q^{t+1} \log p(x, z|\theta)$$

In [Dempster et al., 1977b], Dempster et al. provide the proof of the convergence property of EM algorithm. Also, justified by the equations above, the monotonic increasing objective function gives rise to the sub-optimal solution of the optimization problem.

The algorithm under our framework proposed in this chapter makes use of a graphical model to model the dependency between the random variables of our model and Expectation Maximization (EM) algorithm to learn the parameters.

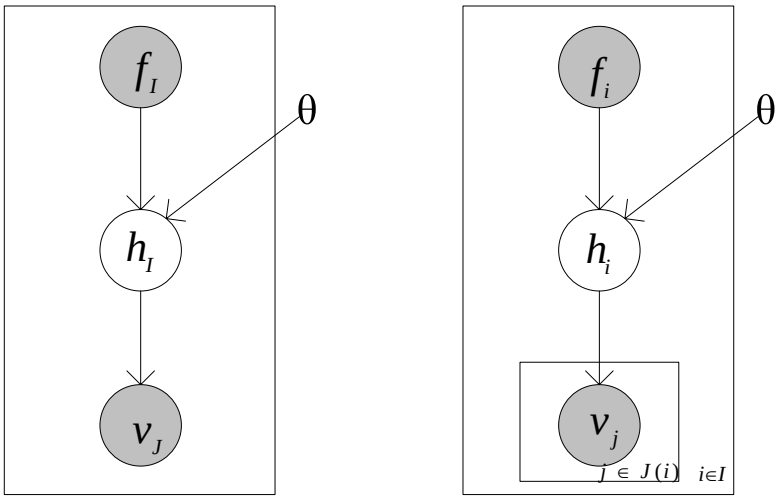
4.3.3 A Graphical Model Fitting with Our Framework

Figure 4.2 provides the graphical representations of the consumer-generated review helpfulness model. f_I denotes the feature set of consumer-generated reviews, h_I denotes the set of helpfulness, and v_J denotes the votes submitted by the readers. We suppose that each feature appears independently in a review document. Then, h_i can be calculated by the following linear summation model:

$$h_i = f_i A^T + z, \tag{4.3.1}$$

where the variances of the Gaussian noise term $z \sim \mathcal{N}(0, \sigma^2)$.

Let us denote the parameters A and σ^2 by θ. Then the helpfulness distribution can be written in the form of $p_{\mathbf{H}|\mathbf{F}}(h_i|f_i; \theta)$. Consider the generative process for a



(a) Overall graphical model representation

(b) Detailed graphical model representation

Figure 4.2: Graphical model representation

consumer-generated review containing features f_i, the probability density function of h_i is:

$$p_{\mathbf{H}|\mathbf{F}}(h_i|f_i;\theta) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(h_i - f_i A^T)^2}{2\sigma^2}). \tag{4.3.2}$$

From the graphical model specified in Figure 4.2(b), it can be easily discovered that:

$$p_{\mathbf{H}|\mathbf{V}}(h_i, v_{J(i)}|f_i) = p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i) p_{\mathbf{H}|\mathbf{F}}(h_i|f_i)$$

and

$$p_{\mathbf{V}|\mathbf{F}}(v_{J(i)}|f_i) = \int_{h_i} p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i) p_{\mathbf{H}|\mathbf{F}}(h_i|f_i) d_{h_i}.$$

By the integration over the continuous hidden variable h_I, we obtain the marginal distribution in the form of

$$p_{\mathbf{V}|\mathbf{F}}(v_J|f_I) = \int_{h_I} \prod_{i \in I} p_{\mathbf{H}|\mathbf{F}}(h_i|f_i;\theta) \prod_{j \in J(i)} p_{\mathbf{V}|\mathbf{H}}(v_j|h_i) d_{h_I}, \tag{4.3.3}$$

Derived from Figure 4.2(a) and Figure 4.2(b), and taking the product of the marginal probabilities of the helpfulness of single document, we obtain the probability of observing all documents' helpfulness collectively given the features of all document and a set of parameters:

$$p_{\mathbf{H}|\mathbf{F}}(h_I|f_I;\theta) = \prod_{i\in I} p_{\mathbf{H}|\mathbf{F}}(h_i|f_i;\theta). \tag{4.3.4}$$

Given the helpfulness of h_I, the distribution of $v_{J(i)}$ is given by:

$$p_{\mathbf{V}|\mathbf{H}}(v_J|h_I) = \prod_{i\in I} p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i). \tag{4.3.5}$$

Considering the conditional joint distribution of votes v_J and helpfulness h_I, we have:

$$\begin{aligned} p_{\mathbf{V},\mathbf{H}|\mathbf{F}}(v_J,h_I|f_I;\theta) &= p_{\mathbf{H}|\mathbf{F}}(h_I|f_I;\theta)p_{\mathbf{V}|\mathbf{H},\mathbf{F}}(v_J|h_I,f_I;\theta) \\ &= p_{\mathbf{H}|\mathbf{F}}(h_I|f_I;\theta)p_{\mathbf{V}|\mathbf{H}}(v_J|h_I) \\ &= \left[\prod_{i\in I} p_{\mathbf{H}|\mathbf{F}}(h_i|f_i;\theta)\right]\left[\prod_{i\in I} p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i)\right] \\ &= \prod_{i\in I} p_{\mathbf{H}|F}(h_i|f_i;\theta)p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i). \end{aligned} \tag{4.3.6}$$

Based on Eq. (4.3.4), Eq. (4.3.5) and Eq. (4.3.6), for a given value of parameters θ and features f_I, the distribution of v_J is then obtained by integrating over h_I:

$$\begin{aligned} p_{\mathbf{V}|\mathbf{F}}(v_J|f_I;\theta) &= \int_{h_I} p_{\mathbf{V},\mathbf{H}|\mathbf{F}}(v_J,h_I|f_I;\theta)dh_I \\ &= \int_{h_I} \prod_{i\in I} p_{\mathbf{H}|\mathbf{F}}(h_i|f_i;\theta)p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i)dh_I \\ &= \prod_{i\in I} \int_{h_i} p_{\mathbf{H}|\mathbf{F}}(h_i|f_i;\theta)p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i)dh_i \end{aligned} \tag{4.3.7}$$

Our goal in adapting the feature vectors is to obtain a model of the helpfulness distribution of consumer-generated reviews. As can be seen from the discussion above,

the optimization problem leads to the following maximization problem:

$$\begin{aligned}\hat{\theta} &= \operatorname*{argmax}_{\theta} p_{\mathbf{V}|\mathbf{F}}(v_J|f_I;\theta) \\ &= \operatorname*{argmax}_{\theta} \log p_{\mathbf{V}|\mathbf{F}}(v_J|f_I;\theta) \\ &= \operatorname*{argmax}_{\theta} \sum_{i\in I} \log \int_{h_i} p_{\mathbf{H}|\mathbf{F}}(h_i|f_i;\theta) p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i) dh_i \end{aligned} \tag{4.3.8}$$

In other words, we need to find θ which maximize

$$\sum_{i\in I} \log \int_{h_i} g_1(h_i, f_i, \theta) \prod_{j\in J(i)} g_2(v_j, h_i) dh_i.$$

There is no close form solution for $\hat{\theta}$. The function $g_1(\alpha_i, f_i, \theta)$ is concave in θ, so the problem is a concave optimization problem.

By the optimization, we can find the $\hat{\theta}$ which locally maximizes $p_{\mathbf{V}|\mathbf{F}}(v_J|f_I;\theta)$ by the coordinate ascent method. In this work, we apply EM algorithm to estimate the parameters of the entire model.

4.3.4 Learning Algorithm

We make use of Expectation Maximization(EM) algorithm [Dempster et al., 1977a] to solve the optimization problem to find the $\hat{\theta}$ which maximizes $p_{\mathbf{V}|\mathbf{F}}(v_J|f_I;\theta)$ for our proposed model. In variational inference for h, each coordinate can be optimized analytically. However, Maximum Likelihood is required when optimizing for θ.

This iterative optimization method can find a local maximum of the likelihood of given training data according to the parameters of our probabilistic model. It is easy to show that:

$$q^{t+1}(h_I) = \operatorname*{argmax}_{q} \mathcal{L}(q, \theta^t) = p_{\mathbf{H}|\mathbf{F},\mathbf{V}}(h_I|f_I, v_J;\theta^t)$$

and:

$$\theta^{t+1} = \operatorname*{argmax}_{\theta} \int_{h_I} q^{t+1}(h_I) \log p_{\mathbf{V},\mathbf{H}|\mathbf{F}}(v_J, h_I|f_I;\theta) dh_I.$$

The following statements show how to do the maximization of estimating the parameters.

E-Step:

$$\begin{aligned} q^{t+1}(h_I) &= p_{\mathbf{H}|\mathbf{F},\mathbf{V}}(h_I|f_I, v_J; \theta^t) \\ &= \prod_{i \in I} p_{\mathbf{H}|\mathbf{F},\mathbf{V}}(h_i|f_i, v_{J(i)}; \theta^t) \end{aligned} \tag{4.3.9}$$

We denote $p_{\mathbf{H}|\mathbf{F},\mathbf{V}}(h_i|f_i, v_{J(i)}; \theta^t) := q_i(h_i)$, Then, $q^{t+1}(h_I)$ is completely specified by $\{q_i(h_i) : i \in I\}$.

$$\begin{aligned} q_i^{t+1}(\alpha_i) &= \frac{p_{\mathbf{H},\mathbf{V}|\mathbf{F}}(h_i, v_{J(i)}|f_i; \theta^t)}{p_{\mathbf{V}|\mathbf{F}}(v_{J(i)}|f_i; \theta^t)} \\ &\propto p_{\mathbf{H}_i,\mathbf{V}_{J(i)}|\mathbf{F}_i}(h_i, v_{J(i)}|f_i; \theta^t) \\ &= p_{\mathbf{H}|\mathbf{F}}(h_i|f_i, \theta^t) p_{\mathbf{V}|\mathbf{H},\mathbf{V}}(v_{J(i)}|h_i, v_i, \theta^t) \\ &= p_{\mathbf{H}|\mathbf{F}}(h_i|f_i; \theta^t) p_{\mathbf{V}|\mathbf{H}}(v_{J(i)}|h_i) \\ &= g_1(h_i, f_i, \theta^t) \prod_{j \in J(i)} g_2(v_j, h_i) \end{aligned} \tag{4.3.10}$$

M-step:

$$\begin{aligned} \theta^{t+1} &= \operatorname*{argmax}_{\theta} \int_{h_I} q^{t+1}(h_I) \log p_{\mathbf{V},\mathbf{H}|\mathbf{F}}(v_J, h_I|f_I; \theta) dh_I \\ &= \operatorname*{argmax}_{\theta} \int_{h_I} (\prod_{i \in I} q_i^{t+1}(h_i)) \log \Big[\prod_{i \in I} \big[g_1(h_i, f_i; \theta) \prod_{j \in J(i)} g_2(v_j, h_i) \big] \Big] dh_I \\ &= \operatorname*{argmax}_{\theta} \int_{h_I} (\prod_{i \in I} q_i^{t+1}(h_i)) \sum_{i \in I} \Big[\log \big[g_1(h_i, f_i, \theta) \prod_{j \in J(i)} g_2(v_j, h_i) \big] \Big] d\alpha_I \\ &= \operatorname*{argmax}_{\theta} \int_{h_I} (\prod_{i \in I} q_i^{t+1}(h_i)) \sum_{i \in I} [\log g_1(h_i, f_i, \theta) + \sum_{j \in J(i)} \log g_2(v_j, h_i)] dh_I \end{aligned}$$

$$
\begin{aligned}
&= \underset{\theta}{\operatorname{argmax}} \Bigg[\int_{h_I} [\prod_{i \in I} q_i^{t+1}(h_i)] \sum_{i \in I} \log g_1(h_i, f_i, \theta) dh_i \\
&+ \int_{h_I} [\prod_{i \in I} q_i^{t+1}(h_i)] \sum_{i \in I} \sum_{j \in J(i)} \log g_2(v_j, h_i) dh_I \Bigg] \\
&= \underset{\theta}{\operatorname{argmax}} \int_{h_I} [\prod_{i \in I} q_i^{t+1}(h_i)] \sum_{i \in I} \log f(h_i, f_i, \theta) dh_I \\
&= \underset{\theta}{\operatorname{argmax}} \sum_{i \in I} \int_{h_i} q_i^{t+1}(h_i) \log f(h_i, f_i, \theta) dh_i \\
&= \underset{\theta}{\operatorname{argmax}} \sum_{i \in I} \int_{h_i} q_i^{t+1}(h_i)(h_i - f_i A^T) dh_i. \qquad (4.3.11)
\end{aligned}
$$

In summary, the procedure for our EM algorithm consists of the following steps:

1. Initialize $\theta^t = \theta^0$, $\theta^{t-1} = \theta^0$, where θ^0 is the initial estimate of the parameter and the superscript represents the index of iteration. For helpfulness prediction, we set $\theta^0 = 1$

2. Calculate $q_i^t(h_i)$ by following Eq.(4.3.10) with respect to θ^{t-1}.

3. Update $q_i^t(h_i)$ by Eq.(4.3.9).

4. Update θ^t by Eq.(4.3.11).

5. Set $t = t + 1$. If convergence condition is not satisfied, go to step 2.

Note that in the M-step of the algorithm, we fixed σ for each iteration and then σ can be deducted during calculating the maximum θ. For an evaluation at algorithm level, we choose the most probable σ for each consumer-generated review, namely, to find a $\hat{\theta}$ to maximize the log-likelihood of $p_{\mathbf{V}|\mathbf{F}}(v_J|f_I;\theta)$. We predefine a σ candidate set of $\{0.1, 0.2, 0.3, \ldots, 1.0\}$ and iteratively go through this algorithm. As a result, 10 pairs of σ, A is learned by the EM algorithm.

As discussed before, the parameter σ can be globally optimized by 4.2.2. This forces the width of the distribution to be the same for every document. Therefore, we

choose the σ and its corresponding A which maximize the probability of observing the training data. When a new consumer-generated review content arrives, the parameter θ which maximizes $\log p_{\mathbf{V}|\mathbf{F}}(v_{J(i)}|f_i;\theta)$ is chosen to calculate the helpfulness distribution of this consumer-generated review content. Afterwards, the helpfulness bias of consumer-generated reviews will be calculated and consumer-generated reviews are ranked according to their corresponding helpfulness bias.

4.4 Experimental Results

To demonstrate the effectiveness of the our proposed approach, we apply our probabilistic approach to the real consumer-generated review set, in form of online product reviews, obtained from Amazon.com and compare it with the most commonly used machine learning method Support Vector Regression (SVR) and Artificial Neural Networks (ANN). This section also presents our method of evaluation, experimental setups.

4.4.1 Dataset

Our experiments explicitly focus on the categories of electronic products and books. We crawled 1002 HDTV reviews, 1492 camera reviews, and 2408 book reviews from the website of *Amazon* [2] at August, 2009. These reviews [3] have been evaluated by at least 10 consumers as helpful or not helpful. We utilized the bag-of-word approach to build the language model. Each feature is a "non-stop, stemmed word" [4] and the value of this feature is a boolean value of the occurrence of the word on the review.

[2] `http://www.amazon.com`

[3] The complete dataset is available at our web set at `http://www.site.uottawa.ca/~rzhan025/helpfulness.html`

[4] "stop words" mean common words that provide no real descriptive content; "stemmed words" mean words have been normalized to their roots (base words), for example, "products" and "production" will be normalized to "produc". "non-stop, stemmed word" means the word is not a stop word and it also has been stemmed. From now on, whenever we use the term word, we mean "non-stop, stemmed word".

We make use of the vocabulary of "word" or "terms" as the feature set of the model to build a document-term matrix. Each the matrix has been normalized to zero-mean.

There are more than 15,000 identical terms in each of the consumer-generated review corpus, so the term-document matrix of the document set is sparse and we need to reduce the document dimensions. After normalizing the document-term matrix, we performed principal component analysis (PCA) [Jolliffe, 2002] to reduce the dimension of term-document matrix. Therefore the top 200 components, which dominate about 70 percent of the total variance, are selected in the later on experimentation. We compute the principle eigenvectors of the covariance matrix for each training set and project the original document-term vector to a 200-dimensional space.

4.4.2 Results and Analysis

Evaluation of the Probabilistic Helpfulness Metric

First, we investigate the advantages of making use of the proposed probabilistic helpfulness metric as the learning target. To evaluate the performance of our proposed probabilistic helpfulness metric, we compare the rank correlation coefficient resulting from SVR with the conventional helpfulness metric and the probabilistic helpfulness metric.

The SVR is a commonly used machine-learning algorithm successfully used for assessing review helpfulness [Kim et al., 2006; Zhang and Varadarajan, 2006]. The SVR is applied in this study to compare the effectiveness with our probabilistic helpfulness metric. In this evaluation process, we conduct the following two experiments:

- SVR with positive vote fraction as helpfulness metric, which will be denoted by SVR-A
- SVR with proposed probabilistic helpfulness metric, which will be denoted by SVR-B

In order to evaluate the performance of our proposed helpfulness metric on "words of few mouths", we construct a "few-vote" dataset from the whole collected data set by randomly selecting k (a small number) user's votes for each review and removing all other votes. In our study, for the diversity of the generated data, we choose a small number $k = 5$ uniformly, at random for each review when building the few-vote data set. Noting that the few-vote data set and the many-vote data set contain the same collection of reviews and that their difference is that in the many-vote data set, each review is voted by no fewer than M users and in the few-vote dataset, each review is voted by k users.

We partition the set of reviews into training reviews and testing reviews, where 2/3 of the reviews are training reviews and 1/3 are testing reviews. The partitioning is performed repeatedly using random sub-sampling, namely, that a random 1/3 fraction of the reviews are selected as testing reviews and the remaining 2/3 are selected as training reviews. A total of 50 random partitions are generated for this study.

After each experiment using any algorithm, the testing reviews are ranked according to their predicted helpfulness metric. Then Spearman's rank correlation coefficient ρ between the helpfulness ranks of these reviews and the corresponding helpfulness ranks obtained by comparing the positive vote fractions of these reviews the many-vote dataset is computed. The definition of ρ is given below.

$$\rho = 1 - \frac{6 \sum_{i \in \mathcal{T}} (x_i - y_i)^2}{|\mathcal{T}|(|\mathcal{T}|^2 - 1)}, \tag{4.4.1}$$

where x_i is the rank of review i according to the helpfulness metric predicted by an algorithm and y_i is the rank of review i according to the positive vote fraction of review i obtained from the many-vote dataset.

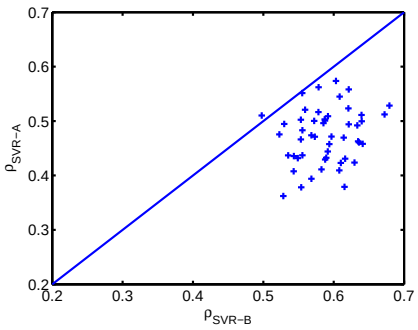
For simplicity, we refer to ρ as the helpfulness ranking correlation and will use it as the main performance measure.

The average $\bar{\rho}$ of helpfulness correlations may be computed across all random

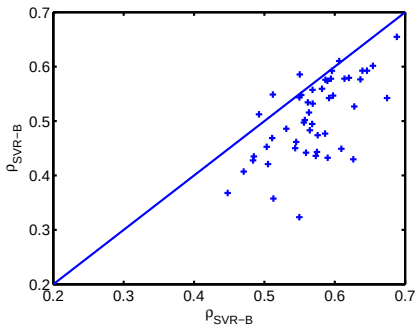
partitions to obtain the overall performance of an algorithm. In addition, the correlation values can be used in a t-test to determine whether an algorithm, say $A1$, performs significantly differently from another algorithm, say $A2$. For example, if helpfulness correlation values are used as the statistics for the test, the t-value is defined as

$$t := \frac{\bar{\rho}(A1) - \bar{\rho}(A2)}{S}, \tag{4.4.2}$$

where S is the standard deviation of $\rho(A1) - \rho(A2)$. The p-value may be computed from the t-value using the student-t distribution and serve as a measure of significance (the lower the p-value, the higher the significance, commonly accepted p-value being lower than 0.05). Similar t-test may be carried out using helpfulness rank correlation as the test statistics.



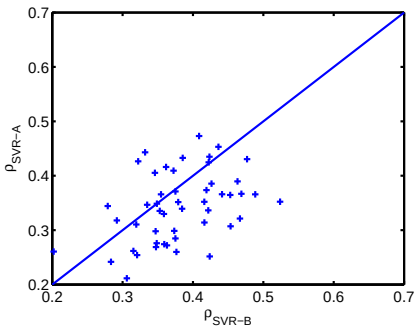
(a) SVR-A vs SVR-B (HDTV many-vote dataset): $\bar{\rho}_{SVR-B}$=0.588, $\bar{\rho}_{SVR-A}$=0.472, p-value=3.67e-19



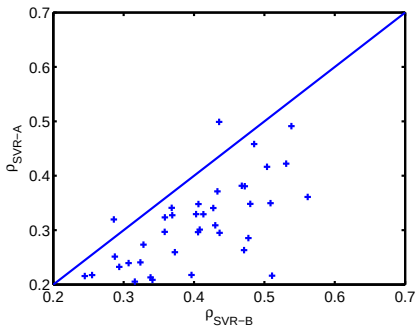
(b) SVR-A vs SVR-B (HDTV few-vote dataset): $\bar{\rho}_{SVR-B}$=0.567, $\bar{\rho}_{SVR-A}$=0.506, p-value=2.15e-10

Figure 4.3: Comparison of helpfulness rank correlation using SVR-A and SVR-B on HDTV reviews.

Figure 4.3, 4.4 and 4.5 shows a set of scatter plots that compare the helpfulness correlation between SVR-A, and SVR-B for HDTV, camera and book data in few-vote and many-vote experiments. Each point in any plot corresponds to one result of a single run of the experimentation. It is visually apparent that the points in each of these plots primarily scatter below the $y = x$ diagonal line, suggesting that there

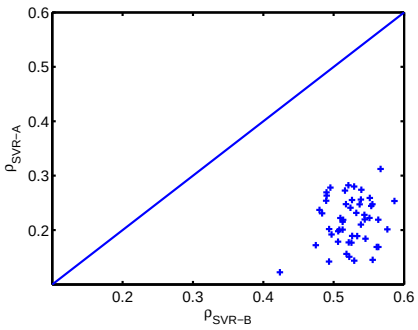


(a) SVR-A vs SVR-B (Camera many-vote dataset): $\bar{\rho}_{SVR-B}$=0.376, $\bar{\rho}_{SVR-A}$=0.344, p-value=0.007

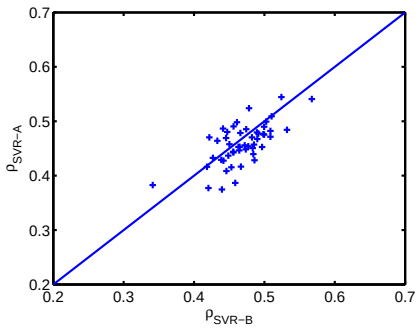


(b) SVR-A vs SVR-B (Camera few-vote dataset): $\bar{\rho}_{SVR-B}$=0.392, $\bar{\rho}_{SVR-A}$=0.27, p-value=2.82e-11

Figure 4.4: Comparison of helpfulness rank correlation using SVR-A and SVR-B on camera reviews.



(a) SVR-A vs SVR-B (Book many-vote dataset): $\bar{\rho}_{SVR-B}$=0.52381, $\bar{\rho}_{SVR-A}$=0.21703, p-value=1.57e-41



(b) SVR-A vs SVR-B (Book few-vote dataset): $\bar{\rho}_{SVR-B}$=0.469, $\bar{\rho}_{SVR-A}$=0.458, p-value=0.001

Figure 4.5: Comparison of helpfulness rank correlation using SVR-A and SVR-B on book reviews.

is a significant performance advantage of SVR-B over SVR-A.

We also perform the significance tests for the rank correlation coefficient to evaluate if there is a statistically significant difference in the correlation coefficients resulting from SVR-A and SVR-B. The p-values of the t-test are all smaller than 0.005. These results confirm that our proposed probabilistic metric is more suitable for the

learning purpose, even when only a small number of votes available in the training set.

Since the proposed helpfulness metric outperforms the positive vote fraction, we will use our helpfulness metric as the learning target of the machine learning approaches that we compare with in the following experiments.

Experiment Setup

To validate the ranking performance of our model, we compare our probabilistic model with Support Vector Regression (SVR) [Burges, 1998], Artificial Neural Network (ANN) [Bishop, 1996], and Linear Regression.

As mentioned above, the SVR is a commonly used machine-learning algorithm successfully used for assessing review helpfulness [Kim et al., 2006; Zhang and Varadarajan, 2006]. We use the LibSVM [Chang and Lin, 2001] tool to perform the Support Vector Regression. The parameters of the SVR, C and g, were chosen by applying a 10-fold cross-validation and a grid search on a logarithmic scale. A linear regression model has also been used for scoring the utility of reviews in [Zhang and Varadarajan, 2006]. Basically, linear regression is to find parameters which minimize the sum of the square deviations of all observed data. We applied multiple linear regression model to predict the helpfulness of reviews and examined the ranking performance of this technique.

The ANN is one of the most popular machine-learning algorithms to solve modeling and predicting problems. We implement a three-layer error back-propagation (BP) ANN in this study. The number of neurons in its hidden layer is chosen to be 10. Each node utilizes sigmoid transfer function. The output node makes use of log-sigmoid transfer function. The training progress is set to be stopped after 1000 iterations of learning, or when present error value less than 0.001 is reached.

For all of these models, we use the PCA projected review data as the training

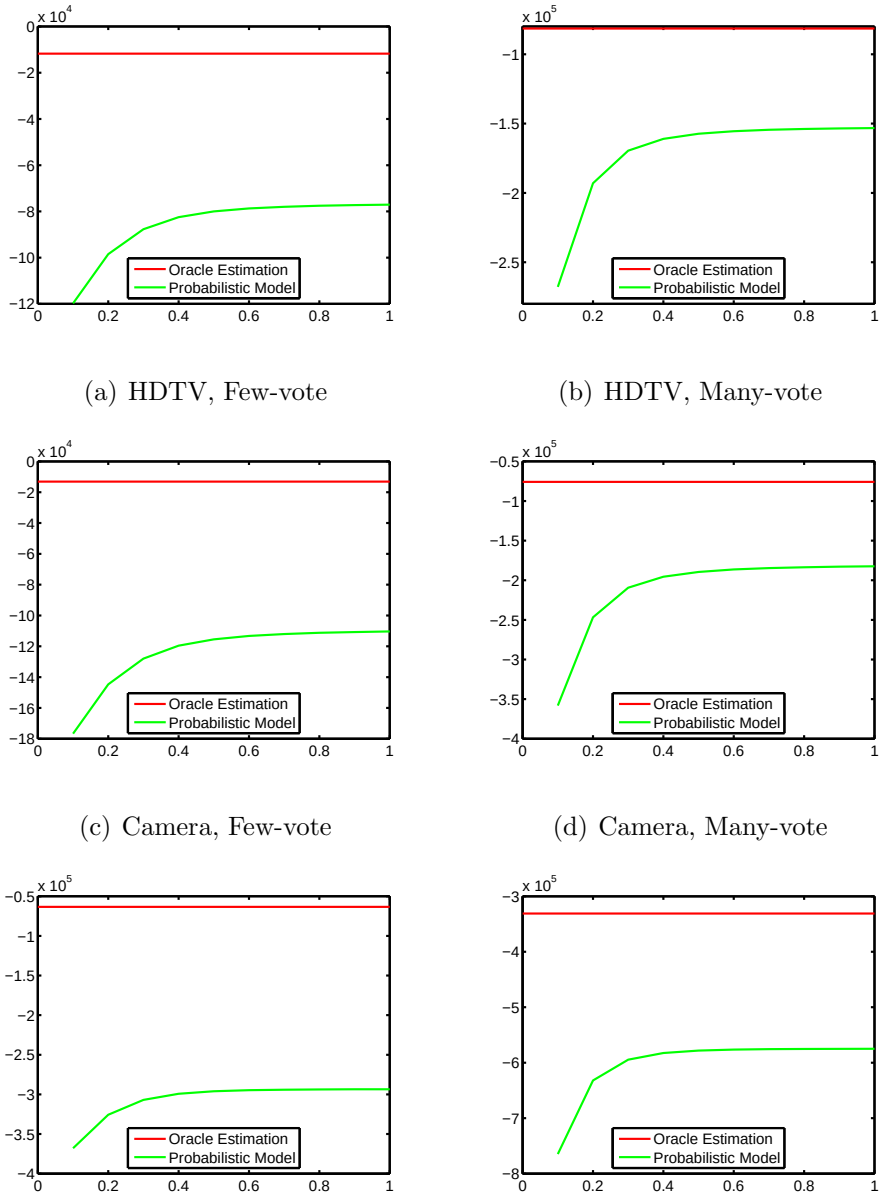
data and test data. The learning target for these algorithms is the helpfulness bias of the "Oracle". The helpfulness predictions from these models are evaluated by computing and comparing the Spearman's Rank Correlation Coefficient between the predicted rankings and the "Oracle" rankings.

In order to investigate the effectiveness of models towards consumer-generated reviews with different voters population sizes, we separate each categories of documents into datasets according to the number of voters associated with each document. "Many-vote" set contains all consumer-generated reviews and the corresponding voter opinions. We randomly choose 5 opinions for each consumer-generated review from "many-vote" set to form "few-vote" set.

To obtain a fair comparison, we perform 10-fold cross-validation on our probabilistic model, SVR, ANN, and linear regression algorithm and compare the cross-validation performances.

Goodness-of-Fit Tests

Figure 4.6 demonstrates the goodness-of-fit of the "Oracle" (straight line) and our model (curve) with varying σ from 0.1 to 1 with the training data. As described in our proposed EM algorithm, the σ which maximizes the likelihood of model and the corresponding parameter set θ is chosen to be the final parameter to build our helpfulness model. We make use of a Gaussian distribution to model the relationship between h_i and f_i. With this Gaussian distribution, if a document is associated with a small number of votes, the σ will be greater than the one with large number of votes. If σ is selected as a large number, the distribution of $p_{\mathbf{H}|\mathbf{F}}(h_i|f_i;\theta)$ will run out of the range of (0,1). Therefore, in practice, one more consideration in choosing appropriate σ is desired. The value of σ can not be too big, so that most of the Gaussian distribution stays between (0,1). In addition, based on the knowledge of the training dataset (as illustrated in Figure 4.6), σ can not be too small to maximize the likelihood



(a) HDTV, Few-vote

(b) HDTV, Many-vote

(c) Camera, Few-vote

(d) Camera, Many-vote

(e) Books, Few-vote

(f) Books, Many-vote

Figure 4.6: The goodness-of-fit of our model and "Oracle" on the training data.

of our model. We examine our model with three categories of consumer-generated reviews by varying σ from 0.1 to 1 and find that a value of $\sigma = 0.2$ is a satisfactory choice that meets these requirements and is used for the following experimentation.

Table 4.2: The goodness-of-fit of our probabilistic model, SVR, ANN and linear regression for the testing data (10-fold cross-validation). "Few-vote" and "many-vote" dataset contain review documents which have been voted by 5 voters and more than 10 voters respectively.

Online Reviews	Algorithm	Few-vote	Many-vote
HDTV	Oracle	-9096	-9096
	Probabilistic Model	-14155	-13013
	SVR	-19194	-18757
	BP-ANN	-19960	-22841
	Linear Regression	-15602	-16072
Camera	Oracle	-8513	-8513
	Probabilistic Model	-16880	-15723
	SVR	-19584	-19101
	BP-ANN	-17541	-18774
	Linear Regression	-19656	-19574
Books	Oracle	-36814	-36814
	Probabilistic Model	-43777	-43621
	SVR	-47447	-47076
	BP-ANN	-52223	-127160
	Linear Regression	-54467	-57468

Table 4.2 demonstrates the difference of the log-likelihood between the "Oracle", our algorithm, SVR, linear regression and ANN with the testing data. It can be seen that the goodness-of-fit of our model consistently outperforms SVR, ANN and linear regression for reviews for all the three categories and both datasets of consumer-generated reviews. This result indicates that our model fits the observed data better than other algorithms no matter the available voter population for the

training purpose is big or small. The log-likelihood of the prediction results maybe not perfect for evaluating the effectiveness of non-probabilistic algorithms. Therefore, we introduce the rank correlation coefficient to evaluate the ranking performance of different algorithms, probabilistic and non-probabilistic.

Table 4.3: Spearman's rank correlation of our probabilistic model, SVM Regression, ANN and linear regression (10-fold cross-validation). "Few-vote" and "many-vote" datasets contain review documents which have been voted by 5 voters and more than 10 voters respectively.

Online Reviews	Algorithm	Few-vote	Many-vote
HDTV	Probabilistic Model	0.62	0.65
	SVR	0.56	0.58
	BP-ANN	0.54	0.57
	Linear Regression	0.50	0.54
Camera	Probabilistic Model	0.47	0.49
	SVR	0.41	0.40
	BP-ANN	0.44	0.46
	Linear Regression	0.40	0.40
Books	Probabilistic Model	0.50	0.51
	SVR	0.42	0.47
	BP-ANN	0.39	0.45
	Linear Regression	0.42	0.46

Ranking Performance

In order to compare the obtained helpfulness bias from different learning algorithms with the helpfulness bias of "Oracle", we evaluate the prediction accuracy of helpful rankings in terms of Spearman's ranking correlation coefficient which has been defined

in Eq.(4.4.2).

The ranking performance of different approaches is analyzed by 10-fold cross-validation. Table 4.3 shows the experimental results of our probabilistic model, SVR, ANN, and Linear Regression algorithm on HDTV, camera and book reviews. The results demonstrate that the effectiveness of our probabilistic model and indicate that our model consistently outperforms SVR, ANN and linear regression for reviews with any voter population size.

For the reviews from all three review categories, compared with the other existing approaches, a significant improvement of the Spearman's rank correlation coefficient is obtained by our model. In each dataset, the results show a close correlation between the "Oracle" and our probabilistic model. For instance, the correlation coefficients of our model and the "Oracle" for each set of reviews are 0.62 and 0.65 respectively for HDTV reviews, 0.47 and 0.49 for camera reviews, and 0.50 and 0.51 for book reviews. This result shows that our probabilistic model has a significant performance advantage over SVR, ANN and linear regression.

Furthermore, as the rank correlation coefficients listed are averaged values of the 10 runs of experimentations, we perform t-test to evaluate the difference between the 10 resulted ranking correlations by our algorithm and other algorithms' ranking correlations of 10 runs. We find that the difference is significant at the 0.05 probabilistic level. These results strongly indicate that our model can consistently learn the helpfulness model under any voter population size and outperforms than SVM Regression, ANN and linear regression.

As we expected, SVR, ANN and Linear Regression with the reviews that are voted by more than 10 voters for all the two types of reviews performs better than the reviews voted by 5 voters. This is because that the confidence of the point prediction is low when voter population size is small. When more votes are available for a consumer-generated review, the benchmark helpfulness value, which is used for training the point estimate model, is more certain and the ranking performance is

improved by the increased certainty of data. Whereas our probabilistic approach, that estimates the distribution of the voter opinion and estimate the helpfulness bias based on the estimated helpfulness distribution, is less likely to be affected by the shortage of voter opinions.

4.5 Discussion and Conclusion

In this chapter we have investigated the disadvantage of the conventional helpfulness definition, particularly in the case of "words of few mouths", and proposed a probabilistic framework to analyze the helpfulness distribution of online product reviews.

The main contribution of this study is a novel probabilistic framework for modeling the helpfulness of consumer-generated reviews. Methodologically, we first create a probabilistic framework, or a family of hypotheses, to characterize the statistical dependency between "items", users, and their opinions. Such a model is typically characterized by a set of parameters, and some of these parameters or their derived quantities are made to reflect the objective of the recommender system. We then select a parameter setting of the model, or a single hypothesis, that "best" explains the available dataset in some well-principled sense of optimality.

This probabilistic framework provides a theoretical interpretation and a mathematical estimation technique to model the helpfulness distribution of consumer-generated review. We have utilized the graphical modeling and EM algorithm to build a helpfulness assessment model under the proposed framework. Both of the mathematical inference and experimental results validate our framework. Our framework is clearly explained by the helpfulness inference. We experimentally compare the probabilistic method with the Support Vector Regression (SVR) method and artificial neural network (ANN), the current state of the art for such applications, and demonstrate the superior performance of the probabilistic method. the experimental results show that a model under our framework can efficiently rank online reviews as

our algorithm outperforms the existing helpfulness assessment approaches by a clear margin.

We have also proposed a probabilistic helpfulness metric for the conventional machine learning based predictors. Previous studies simply make use of the positive vote fraction as the benchmark to evaluate the helpfulness of consumer-generated reviews. As this benchmark hardly represents the true helpfulness value in the presence of "words of few mouths" phenomenon, existing approaches only take ones which have been voted by a large number (such as >10) of readers. However, in reality, most of consumer-generated reviews are only associated with a small number of available voter opinions and the votes need time to be collected. The available data for training the helpfulness model will be limited to a small amount of consumer-generated reviews.

Furthermore, the traditional helpfulness definition will treat the fraction of $\frac{9}{10}$, $\frac{18}{20}$, and $\frac{27}{30}$ as the same helpfulness value whereas the confidence of these helpfulness estimates varies a lot depending on voter population size. Our approach solves this problem by estimating the helpfulness distribution instead of a single value. We proposed a helpfulness metric as a benchmark to justify the helpfulness of a consumer-generated review document from the available voter opinions and evaluate the true helpfulness value from the helpfulness distribution. We show that SVR with the proposed probabilistic helpfulness metric outperforms SVR with conventional helpfulness metric (positive vote fraction).

The dynamic nature of the user opinions make it desirable that the recommender systems be capable of adapting itself in real time to the continuously arriving information. This challenges the design of the such recommender systems, particularly when the computation resources allocated to the recommender system are not abundant. The model proposed in this Chapter is unable to handle the online learning problem. In next chapter, we study the problem of designing a real-time predicting model for discovering helpful user-generated reviews.

Chapter 5

Logistic Regression-Based Probabilistic Model

This chapter studies the problem of designing a logistic regression-based probabilistic models that are used to predict whether user-generated contents (i.e. consumer-generated reviews) are helpful based on the readers' opinions (helpful/unhelpful votes) on previous consumer-generated reviews. In some circumstances, an algorithm may not see all the data in advance and decisions must be made make on the basis of partial information about the studied problem instance. In light of this we introduce approaches that incrementally update the parameters of the model. Building on the definition of helpfulness and the proposed framework set forth in Chapter 4, we first develop a logistic regression-based helpfulness prediction model and estimate parameters of this model by a batch (off-line) algorithm. We see that this batch algorithm can be easily extended to an online algorithm and the rate of convergence can be increased by intermittently running batch algorithm on collected data off-line. Experiments on data from Amazon.com suggest that our proposed model in fact outperforms the previously reported prediction algorithms. Then we extend the batch algorithm into an online algorithm to support the real-time helpfulness pre-

diction requirements for recommender system. Finally, an efficient hybrid algorithm is provided to increase the convergence rate and prediction precisions. The online and hybrid algorithm are tested on real-life consumer-generated reviews; experimental results illustrate that the hybrid approach efficiently processes incoming data and generates a reliable helpfulness prediction for users compared to existing approaches.

5.1 Introduction

Existing helpfulness prediction approaches [Kim et al., 2006; Weimer, 2007; Liu et al., 2008] are all off-line algorithms which require complete knowledge of the input. Once the helpfulness model learns from a set of observed consumer-generated reviews and votes, the parameters will not be changed except to rebuild the model. In practice, however, consumer-generated reviews and voter opinions accumulate in a dynamic sense. As additional new consumer-generated reviews become available, there is a need to improve the model based on this incoming data. The computational complexity of existing approaches mentioned above also makes impossible the applicability of implementing a real-time system. In this chapter we focus on designing online algorithms to process incoming consumer-generated reviews incrementally under the helpfulness prediction framework proposed in the previous chapter. We use the probabilistic definition of consumer-generated review helpfulness presented in Chapter 4 which defines the helpfulness of a consumer-generated review as the probability that a random reader will vote the consumer-generated review positively. This definition, intuitively sensible, allows a probabilistic formulation of helpfulness prediction in which the learning machine, instead of attempting to simulate the *functional dependency* of the helpfulness value on a document, attempts to simulate a *probabilistic dependency* between the two quantities.

We first demonstrate this by presenting a batch algorithm similar to a logistic regression algorithm for classification and present its superior performance over the

conventional algorithms. This batch algorithm can be intermittently executed in the hybrid model to increase the rate of convergence. We then implement the logistic regression-based model by an online algorithm and the subsequent experimentation illustrates the applicability of this online algorithm to consumer-generated review streams. Finally, we propose a practical hybrid algorithm for consumer-generated review streams which takes the best of online and batch algorithms and improves the convergence rate and the prediction precision.

The remainder of this chapter is organized as follows: section 2 presents a probabilistic formulation of helpfulness and helpfulness prediction and derives a prediction algorithm resulting from a logistic regression model; section 3 presents an experimental evaluation of our approach. This chapter ends with some discussion and brief conclusions.

5.2 Helpfulness Prediction: Probabilistic Formulation and Logistic Regression

5.2.1 Problem Statement

Within this chapter we focus on designing online helpfulness discovering algorithms that process consumer-generated reviews sequentially as they become available. The problem of incrementally predicting the helpfulness of consumer-generated reviews may be simply phrased as the following problem: given a set of observed consumer-generated reviews and a collection of online readers' votes (in terms of `HELPFUL` or `UNHELPFUL`) on the consumer-generated reviews, determine how "helpful" a new consumer-generated review would be to the users when no reader votes are yet available. At the same time, given an incoming voter opinion, the system should update the parameters of the model.

We formally define the online helpfulness prediction problem as follows. At

a specific time t, we use $d_{I^t} := \{d_i : i \in I^t\}$ to denote the set of all observed consumer-generated reviews where set I^t is a finite indexing set and each d_i, $i \in I^t$ is a consumer-generated review. Similarly, we use $v_{J^t} := \{v_j : j \in J^t\}$ to denote the set of all available votes where set J^t is another finite indexing set and each $v_j, j \in J^t$, is $\{0, 1\}$-valued variable, or a vote, with 1 corresponding to `HELPFUL` and 0 corresponding to `UNHELPFUL`.

The stochastic gradient procedure is the most wildly used online algorithm. We will start by formulating the problem on the whole observed dataset and then extend the problem to be solved by a stochastic gradient procedure. We use I and J to represent all the observed consumer-generated reviews and voter opinions. The association between votes and consumer-generated reviews effectively induces a partition of index set J into disjoint subsets $\{J(i), i \in I\}$, where for each i, $J(i)$ indexes the set of all votes concerning consumer-generated review d_i. In particular, each set $J(i)$ naturally splits into two disjoint subsets $J^+(i)$ and $J^-(i)$, indexing respectively the positive (i.e. `HELPFUL`) votes on consumer-generated review i and the negative (i.e. `UNHELPFUL`) votes on consumer-generated review i.

The helpfulness prediction problem of interest in this chapter can then be rephrased as determining how helpful an arbitrary consumer-generated review d, not necessarily in d_I, would be given d_I, v_J and the partition $\{J(i) : i \in I\}$.

The helpfulness prediction framework proposed in Chapter 4 provides a mathematical formulation of this problem. We will discuss how to build helpfulness prediction by the stochastic approach under our proposed framework in the following section.

5.2.2 Probabilistic Formulation

The helpfulness definition we proposed in Chapter 4 allows for a probabilistic formulation of the helpfulness prediction problem: given the data $(d_I, v_J, \{J(i) : i \in I\})$

generated from the above procedure, determine the distribution $p_{\mathbf{V}|\mathbf{D}}$.

An experienced reader should have identified an apparent similarity between this problem formulation and the probabilistic formulation of standard classification problems in machine learning literature. In classification problems under probabilistic formulation, a classifier can be obtained by determining $p_{\mathbf{V}|\mathbf{D}}$, and each $d \in \mathcal{D}$ is labeled with 1 if $p_{\mathbf{V}|\mathbf{D}}(1|d) > p_{\mathbf{V}|\mathbf{D}}(0|d)$, and labeled with 0 otherwise. There is also, however, a distinction between this problem and the classification problems. In classification problems each "document" in d_I would be associated with only *one* "vote" (or equivalently $|J(i) = 1|$ for all $i \in I$). Nevertheless, the similarity between this problem and classification problems immediately enables rich families of probabilistic classification methodologies and algorithms to be useful or relevant for solving helpfulness prediction problems.

Although one may consider various options when adopting a classification methodology to solve the formulated problem, here we advocate a model-based principled approach. In this approach we first create a family $\Theta_{\mathbf{V}|\mathbf{D}}$ of candidate conditional distributions to the model $p_{\mathbf{V}|\mathbf{D}}$, and then we choose one of the candidates under which the (log)likelihood of the observe data $(d_I, v_J, \{J(i) : i \in I\})$ is maximized. That is, after prescribing the family $\Theta_{\mathbf{V}|\mathbf{D}}$, we solve for

$$p^*_{\mathbf{V}|\mathbf{D}} := \operatorname*{argmax}_{p_{\mathbf{V}|\mathbf{D}} \in \Theta_{\mathbf{V}|\mathbf{D}}} \log p_{V_J|D_I}(v_J|d_I) \tag{5.2.1}$$

Under the assumption specified in the data generation process in which both documents and voting functions are drawn i.i.d., it follows that

$$\begin{aligned} p^*_{\mathbf{V}|\mathbf{D}} &= \operatorname*{argmax}_{p_{\mathbf{V}|\mathbf{D}} \in \Theta_{\mathbf{V}|\mathbf{D}}} \log \prod_{i \in I} \prod_{j \in J(i)} p_{\mathbf{V}|\mathbf{D}}(v_j|d_i) \\ &= \operatorname*{argmax}_{p_{\mathbf{V}|\mathbf{D}} \in \Theta_{\mathbf{V}|\mathbf{D}}} \sum_{i \in I} \sum_{j \in J(i)} \log p_{\mathbf{V}|\mathbf{D}}(v_j|d_i) \end{aligned} \tag{5.2.2}$$

As is common in many machine-learning problems, the huge dimensionality of space $\mathcal{D}$ makes solving problem Eq.(5.2.2) infeasible. A widely-used technique to

reduce the dimensionality is via mapping each document to a low dimensional *feature* vector. Formally, let $\mathcal{F}$ be the image of a given choice of feature generating function $s : \mathcal{D} \to \mathcal{F}$. That is, $\mathcal{F}$ is the space of all feature vectors. The joint distribution $p_{\mathbf{VD}}$ induces a joint distribution $p_{\mathbf{VF}}$ on the cartesian product $\{0, 1\} \times \mathcal{F}$ which further induces a conditional distribution $p_{\mathbf{V}|\mathbf{F}}$ of a random vote $\mathbf{V}$ given a random feature $\mathbf{F}$. The objective of helpfulness prediction as specified in (5.2.2) is then modified to finding

$$p^*_{\mathbf{V}|\mathbf{F}} = \underset{p_{\mathbf{V}|\mathbf{F}} \in \Theta_{\mathbf{V}|\mathbf{F}}}{\operatorname{argmax}} \sum_{i \in I} \sum_{j \in J(i)} \log p_{\mathbf{V}|\mathbf{F}}(v_j | f_i), \tag{5.2.3}$$

where $\Theta_{\mathbf{V}|\mathbf{F}}$ is a family of candidate distributions $p_{\mathbf{V}|\mathbf{F}}$ which we create to model the unknown dependency of $\mathbf{V}$ on $\mathbf{F}$.

At this end, we have not only arrived at a sensible and well-defined notion of helpfulness, we also have translated the problem of helpfulness prediction to an optimization problem. In the remainder of this chapter we present a prediction algorithm similar to the logistic regression algorithm [Hosmer and Lemeshow, 2000] developed in classification literature. The reason of choosing logistic regression algorithm is because that logistic regression can directly model the probability of a voter giving a positive vote for a review document. In addition, logistic function has a range of 0, 1 which also justify the valid use of logistic regression.

5.2.3 Logistic Regression for Helpfulness Prediction

Central to solving the optimization problem specified in Eq.(5.2.3) is the specification of model $\Theta_{\mathbf{V}|\mathbf{F}}$. A good choice of $\Theta_{\mathbf{V}|\mathbf{F}}$ will not only serve to reduce the problem dimensionality yet containing good candidate solutions, but it will also facilitate the development of principled optimization algorithms. Logistic regression model is one of such models.

Using logistic regression we model the probabilistic dependency of $\mathbf{V}$ on $\mathbf{F}$ using

the *logistic function* [Jordan, 1995]. This model is applicable to the helpfulness prediction framework because the output takes value from the range of $(0, 1)$ which can be used to predict the odds ratio of an event.

More precisely, we define

$$p_{\mathbf{V}|\mathbf{F}}(1|f) = \mu(\eta), \tag{5.2.4}$$

where $\mu(\eta)$ is the logistic function defined by

$$\mu(\eta) := \frac{1}{1 + e^{-\eta}},$$

and $\eta := \theta^T f$ for some vector θ having the same dimension as feature vector f [1] . We note that since $p_{\mathbf{V}|\mathbf{F}}(1|f) + p_{\mathbf{V}|\mathbf{F}}(0|f) = 1$, Eq.(5.2.4) completely defines model $\Theta_{\mathbf{V}|\mathbf{F}}$, namely,

$$\Theta_{\mathbf{V}|\mathbf{F}} := \{p_{\mathbf{V}|\mathbf{F}} \text{ satisfying } p_{\mathbf{V}|\mathbf{F}}(1|f) = \frac{1}{1 + e^{-\theta^T f}} : \theta \in \mathbb{R}^m\}, \tag{5.2.5}$$

where we have assumed that each feature vector is m-dimensional.

It is known in the context of binary classification that as long as the conditional distribution of feature given class label is from the exponential family, the conditional distribution of class label given feature is a logistic function. This fact, together with the richness of the exponential family, renders our choice of $\Theta_{\mathbf{V}|\mathbf{F}}$ a robust and general model that is insensitive to the exact form of the distribution governing the dependency between document feature and vote.

For the reader familiar with graphical representation of probability models, Figure 5.1 is a Bayesian network [Neapolitan, 2003] representation of the model.

Now using model $\Theta_{\mathbf{V}|\mathbf{F}}$ defined in Eq.(5.2.5), the optimization problem of Eq.(5.2.3) reduces to solving

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^m}{\operatorname{argmax}} \sum_{i \in I} \sum_{j \in J(i)} \left[v_j \log \mu(f_i) + (1 - v_j) \log \left(1 - \mu(f_i)\right) \right], \tag{5.2.6}$$

[1] In this chapter, all vectors are by default taken as column vectors.

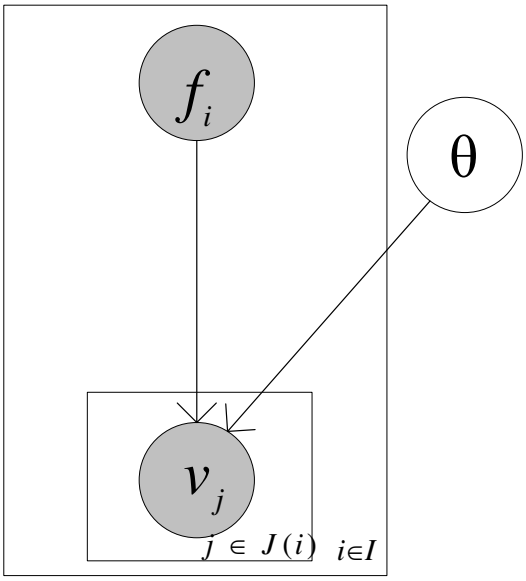


Figure 5.1: Graphical Model Representation

where we have, by a slight abuse of notation, written μ as a function of f, namely, that $\mu(f)$ denotes $\mu(\eta(f))$.

Denoting the objective function in this optimization problem by $l(\theta)$, we have

$$\begin{aligned}
\frac{dl}{d\theta} &= \sum_{i \in I} \sum_{j \in J(i)} [v_j \frac{1}{\mu(f_i)} \frac{d\mu(f_i)}{d\theta} + (1 - v_j) \frac{1}{1 - \mu(f_i)} \frac{d\mu(f_i)}{d\theta}] \\
&= \sum_{i \in I} \sum_{j \in J(i)} [v_j \frac{1}{\mu(f_i)} \mu(f_i)(1 - \mu(f_i)) f_i \\
&+ (1 - v_j) \frac{1}{1 - \mu(f_i)} \mu(f_i)(1 - \mu(f_i)) f_i] \\
&= \sum_{i \in I} \sum_{j \in J(i)} v_j (1 - \mu(f_i)) f_i - \mu(f_i)(1 - v_j) f_i \\
&= \sum_{i \in I} \sum_{j \in J(i)} f_i (v_j - \mu(f_i)) \qquad (5.2.7)
\end{aligned}$$

We will show possible implementations to maximize the likelihood of observed data by making use of the gradient (Eq.(5.2.7)).

5.2.4 Predicting Models

This logistic regression-based model can be implemented as a batch or an online algorithm, where the batch algorithm updates on all existing data simultaneously and the online algorithm updates the estimate as new data is received. As the batch algorithm identifies models using recursive updating and the online algorithm processes one sample at a time, that the computational load of the batch algorithm is clearly higher than the online one and the prediction performance of the online algorithm is surely lower than the batch one. To combine the performance advantage of the batch algorithm and the efficiency advantage of the online algorithm, we then propose a practical hybrid approach which learns online helpfulness model on incoming data and run batch algorithm intermittently on aggregated reviews off-line.

Batch Algorithm

Eq.(5.2.7) allows a gradient ascent algorithm to optimize the objective function in which the value of the objective function can be incrementally increased via updating the configuration of θ according to

$$\theta^{k+1} := \theta^k + \lambda \sum_{i \in I} \sum_{j \in J(i)} f_i(v_j - \mu(f_i)). \tag{5.2.8}$$

where λ is a choice of step size. This update rule leads to a gradual increase in the log-likelihood.

Note that when we take I and J as all the available consumer-generated reviews and voter opinions at a specific time t, namely I^t and J^t. Eq.(5.2.8) then leads to a batch update rule which iteratively updates θ until a predefined equation is achieved.

$$\theta^{t,k+1} := \theta^{t,k} + \lambda \sum_{i \in I^t} \sum_{j \in J^t(i)} f_i(v_j - \mu(f_i)). \tag{5.2.9}$$

where λ is a choice of step size.

We will define a convergence condition so that if the gradient of the objection function becomes small, the parameter is then close to the optimum. We note that in this algorithm we allow the step size λ to decrease gradually. This is due to the fact that a large step size allows a gradient ascent algorithm to hop away from the local optimums but suffers from large oscillation in the value of the objective function, whereas a small step size allows the algorithm to converge with small variation in the value of the objective function but suffers from a higher chance of getting trapped at local optimums.

Online Algorithm

When we consider the arrival of each voter opinion as one time step, Eq.(5.2.8) can be extended to an online gradient ascent algorithm. Each time a new vote on a consumer-generated review arrives, the algorithm takes this incoming consumer-generated review and corresponding available votes to update θ.

Given an existing logistic function-based model which has been learned from existing consumer-generated reviews and votes, and given a new incoming vote, the parameters of the helpfulness model can be updated. We consider the following update rule:

$$\theta^{t+1} = \theta^t + \tilde{\lambda} f_i (v_j^{t+1} - \mu(f_i)), \tag{5.2.10}$$

where v_j^{t+1} is the incoming voter opinion and $\tilde{\lambda}$ is a heuristically chosen small number.

It can be noted that the online algorithms depend solely on current parameters and the incoming data; there is no need for additional information. Moreover, as the simplistic of the update function, this type of algorithm is efficient and can meet the real-time calculation needs.

Hybrid Algorithm

As the online approach only takes into account current incoming voter opinions, it is necessary to intermittently apply a batch algorithm into the online algorithm to collectively infer over the existing available data. As such, the convergence speed and the prediction precision would be increased. We introduce a hybrid approach that takes the best of the above two algorithms. In this hybrid algorithm, online phase and off-line phase perform online and off-line training respectively. In particular, the online algorithm is a light-weight algorithm, which instantaneously processes the newly arrived user opinion, and updates the training results. The off-line algorithm, which implements more sophisticated and computationally costly training algorithms, is only invoked when the system is lightly loaded or has sufficiently computation resources.

The hybrid algorithm can be summarized in this way: at the beginning of each time step t, the online algorithm will process the incoming votes and corresponding consumer-generated reviews between time step $t-1$ and t. If a run batch algorithm condition are met, a batch algorithm will be activated over all of the collected consumer-generated reviews and votes during time step 1 and t. The batch phase can catch a more detailed statistics of the overall helpfulness model and improve the performance of the online prediction. In practice, the criterion can be defined as the system is lightly loaded or has sufficiently computation resources.

We articulate this algorithm as follows:

1. Set ConditionforRunBatchAlgorithm (CRBA)
2. Initialize $t := 0$, set θ^t to a random configuration, and set λ and $\tilde{\lambda}$ to a relatively small value.
3. Set $t := t + 1$.
4. Update θ^t using Eq.(5.2.10).

5. If CRBA is satisfied, update θ^t using Eq.(5.2.9).

where CRBA denotes the batch algorithm running condition.

Note that the initialized θ^t can also be learned from a small set of observed consumer-generated reviews and voter opinions in order to improve the performance.

5.3 Performance Evaluation

5.3.1 Dataset

There is no standard consumer-generated review corpus available. We utilize the web service provided by Amazon.com to collect user-generated reviews. We crawled 23,9705 anonymous voter opinions on 7,665 consumer-generated reviews, which include reviews on HDTVs, cameras, laptops and books. Table 5.1 lists the number of voter opinions and consumer-generated reviews in each category of data. We use the bag of words model to represent text and to build our language model. Each feature is a stemmed word and the value of this feature is a Boolean value of the occurrence of the word on the review. After the parsing and stemming, a document term matrix is returned associated with the helpfulness value of each document.

As in the previous experimentation for batch algorithms, we compute the principle eigenvectors of the covariance matrix for each training set and project the original document term vector to a 200 dimension space.

5.3.2 Evaluation Metrics

Goodness-of-Fit Tests

As discussed in Chapter 4, the objective of our framework is to find a distribution that will maximize the probability of observing the voter opinions. Therefore, our proposed framework naturally provides a methodology for evaluating the fitness of

Table 5.1: The statistics of collected dataset.

Dataset	# of positive votes	# of negative votes	# of reviews
HDTV	20908	6387	751
Camera	55929	8176	1453
Laptop	24274	28145	2193
Book	58913	36977	3268

helpfulness inference algorithms by their likelihood to be helpful. The log-likelihood function is the logarithm of the model formulation. In our case, the fitness of a model/algorithm can be formulated as $\log p_{\mathbf{V}|\mathbf{F}}(v_J|f_I)$, where $p_{\mathbf{V}|\mathbf{F}}(v_J|f_I)$ is the resulted marginal probability distribution learned by an algorithm.

We suppose that it is possible for a learning algorithm M to determine $p_{\mathbf{V}|\mathbf{F}}$ under a particular $p^M_{\mathbf{V}|\mathbf{D}}$, where M can be regarded as a model, either probabilistic or non-probabilistic. In the probabilistic sense M is essentially a set of parameters completely specified the distribution of $p^M_{\mathbf{V}|\mathbf{F}}$. Then for any two algorithms M_1 and M_2, a natural metric to compare their overall performance is to compare the log-likelihood of observation for the resulted model

$$S(M_1, M_2) = \log p^{M1}_{\mathbf{V}|\mathbf{F}} - \log p^{M2}_{\mathbf{V}|\mathbf{F}}.$$

If $S(M_1, M_2) > 0$, then the algorithm $M1$ is better then $M2$. Otherwise, the algorithm $M2$ is better than $M1$.

Ranking Performance

As in Chapter 4, helpfulness rank correlation is used to evaluate the performance of the compared algorithms. Two performance metrics are used to evaluate the performance of the compared algorithms. It is essentially Spearman's rank correlation

coefficient ρ between the helpfulness ranks of the testing reviews predicted by an algorithm and that from the dataset

$$\rho = 1 - \frac{6 \sum_{j \in \mathcal{T}} (x_i - y_i)^2}{|\mathcal{T}|(|\mathcal{T}|^2 - 1)}, \tag{5.3.1}$$

where x_i is the rank of review i according to helpfulness predicted by an algorithm and y_i is the rank of review i according to the helpfulness of review i obtained from the many-vote dataset.

The average $\bar{\rho}$ of helpfulness rank correlations may be computed across all random partitions to obtain the overall performance of an algorithm. In addition, the correlation values can be used in a t-test to determine whether an algorithm, say $A1$, performs significantly different from another algorithm, say $A2$. For example, if the helpfulness correlation values are used as the statistics for the test, the t-value is defined as

$$t := \frac{\bar{\rho}(A1) - \bar{\rho}(A2)}{S}, \tag{5.3.2}$$

where S is the standard deviation of $\rho(A1) - \rho(A2)$. The p-value may be computed from the t-value using the student-t distribution to serve as a measure of significance (the lower the p-value, the higher the significance, commonly accepted p-value being lower than 0.05). Similar t-test may be carried out using the helpfulness rank correlation as the test statistics.

5.3.3 Method of Evaluation

To validate the ranking performance of our proposed models we will consider Support Vector Regression (SVR) [Burges, 1998] and artificial neural network (ANN) for comparison.

The reason we consider SVR algorithms as the primary targets for comparison is because SVR is a widely adopted machine-learning algorithm in many applications and it has also been used for assessing consumer-generated review helpfulness in [Kim

et al., 2006] and [Zhang and Varadarajan, 2006]. We use the LibSVM [Chang and Lin, 2001] tool to perform the Support Vector Regression. The parameters of the SVR, C and g, were chosen by applying a 10-fold cross validation and a grid search on a logarithmic scale.

We implement a three-layer back-propagation (BP) ANN. The number of neurons in the hidden layer is chosen to be 10 and each node utilizes sigmoid transfer function. The output node makes use of a log-sigmoid transfer function and the training of the ANN is terminated after 1000 training iterations or when the error term is less than 0.001.

To evaluate the performance of batch algorithm we partition the set of reviews into the set $\mathcal{N}$ of training reviews and the set $\mathcal{T}$ of testing reviews where 3/4 of the reviews are training reviews and 1/4 are testing reviews. The partitioning is performed repeatedly using random sub-sampling, namely, that a random 1/4 fraction of the reviews are selected as testing reviews and the remaining 3/4 are selected as training reviews. A total of 50 random partitions $(\mathcal{N}, \mathcal{T})$'s are generated in our study.

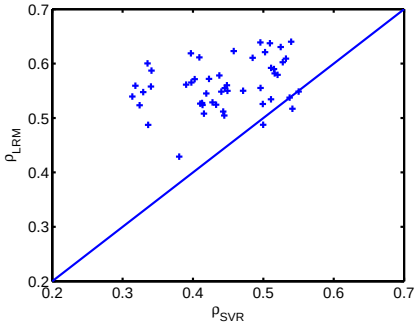
In this setting, two types of experiments may be performed.

Few-Vote Experiment: For each real dataset and each partition $(\mathcal{N}, \mathcal{T})$ of the reviews, we simultaneously train the three algorithms using the training reviews $\mathcal{N}$ where the user votes on these reviews are taken from the few-vote dataset. The trained algorithms are then simultaneously applied to the testing reviews.

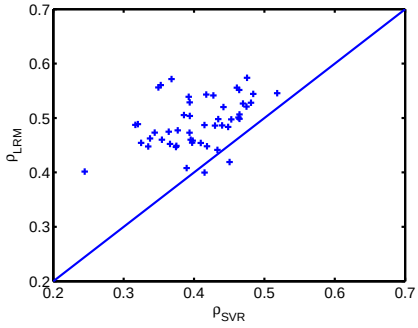
Many-Vote Experiment: A many-vote experiment is identical to the few-vote experiment except that the user votes on the training reviews are taken from the many-vote dataset.

We then randomly choose one of the random partitions $(\mathcal{N}, \mathcal{T})$ as the real-time review accumulating simulation set. We examine the goodness-of-fit and ranking performance of our proposed batch algorithm (bLRM), online algorithm (oLRM) and hybrid algorithm (hLRM) in our experiments.

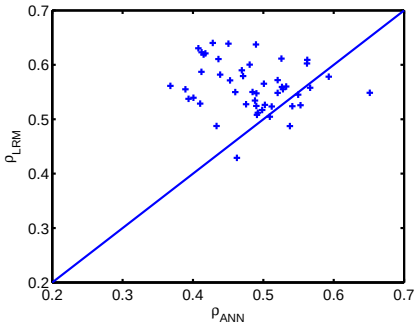
5.3.4 Batch Algorithm Performance



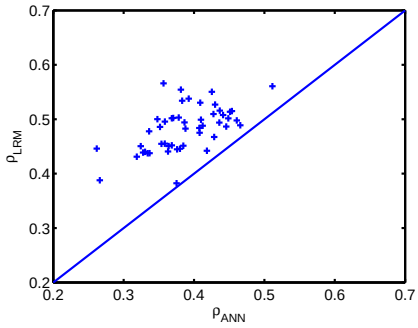
(a) LRM vs SVR (HDTV): $\bar{\rho}_{\text{LRM}} = 0.56073$, $\bar{\rho}_{\text{SVR}} = 0.444$, p-value=5.66e-16



(b) LRM vs SVR (Camera): $\bar{\rho}_{\text{LRM}} = 0.491$, $\bar{\rho}_{\text{SVR}} = 0.406$, p-value=5.84e-15



(c) LRM vs ANN (HDTV): $\bar{\rho}_{\text{LRM}} = 0.561$, $\bar{\rho}_{\text{ANN}} = 0.47204$, p-value=3.95e-5

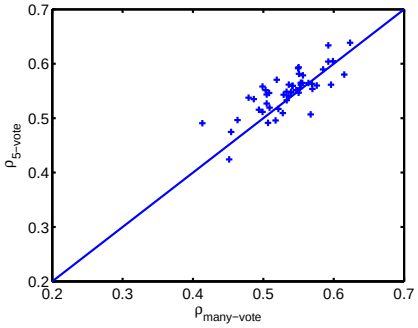


(d) LRM vs ANN (Camera): $\bar{\rho}_{\text{LRM}} = 0.491$, $\bar{\rho}_{\text{ANN}} = 0.351$, p-value=1.25e-13

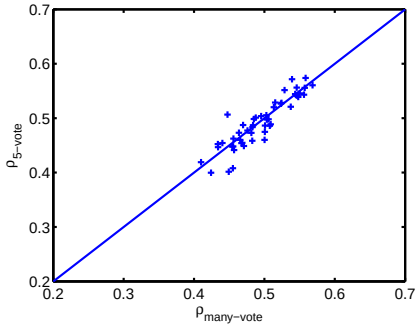
Figure 5.2: Comparison of helpfulness rank correlation using 5-vote datasets. Left column, HDTV dataset; right column camera dataset. Top row, LRM vs SVR; bottom row, LRM vs ANN.

To verify the performance of the batch algorithm we examine only HDTV and camera reviews. Figure 5.2 shows a set of scatter plots that compare the helpfulness rank correlation between the proposed batch algorithm (LRM), SVR, and ANN for HDTV and camera data in few-vote experiments. Each point in any plot corresponds to one partition $(\mathcal{N}, \mathcal{T})$. It is visually apparent that the points in each of these plots primarily scatter above the $y = x$ diagonal line thus suggesting that there is a significant performance advantage of LRM over SVR and ANN. This can also be

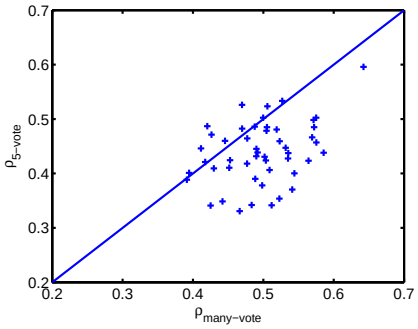
verified by the average of helpfulness rank correlation, $\bar{\rho}$, of the compared algorithms and the p-values of the t-tests (all smaller than 0.005).



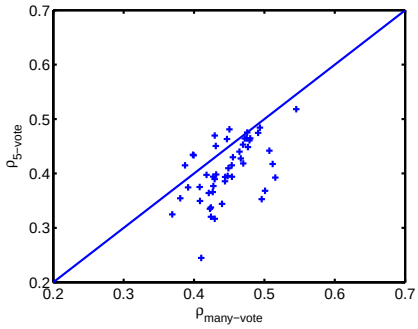
(a) Helpfulness rank correlations of LRM (HDTV): $\bar{\rho}_{\text{many-vote}} = 0.535$,$\bar{\rho}_{5-\text{vote}} = 0.548$



(b) Helpfulness rank correlation of LRM (Camera): $\bar{\rho}_{\text{many-vote}} = 0.494$, $\bar{\rho}_{5-\text{vote}} = 0.491$



(c) Helpfulness rank correlation of ANN (HDTV): $\bar{\rho}_{\text{many-vote}} = 0.497$, $\bar{\rho}_{5-\text{vote}} = 0.428$



(d) Helpfulness rank correlation of SVR (Camera): $\bar{\rho}_{\text{many-vote}} = 0.447$, $\bar{\rho}_{5-\text{vote}} = 0.406$

Figure 5.3: Comparison of the performances of each algorithm between 5-vote data and many-vote data.

Although the proposed LRM algorithm is motivated by the "words of few mouths" phenomenon, nothing in fact would prevent its use as a general helpfulness prediction algorithm even in absence of such a phenomenon. To demonstrate this we also performed many-vote experiments for the same set of random partitions and in fact a similar performance advantage of LRM as those shown in Figure 5.2 is obtained

(data not shown). It is of interest to compile the results obtained in the two set of experiments and to investigate how differently an algorithm performs in few-vote experiments and in many-vote experiments. Figure 5.3 compares the performances of each algorithm between 5-vote data and many-vote data. It can be seen from (a) and (b) that the scattering of the points in the LRM algorithm are tightly around the diagonal line. This indicates that the algorithm is quite robust against the "words of few mouths" phenomenon. In particular, the performance of the algorithm under "words of few mouths" and that in the absence of "words of few mouths" is quite close and this similarity in performance is not only in the average sense but also in the "almost-everywhere" sense. In contrast, as shown in (c) and (d), the performances of SVR and ANN are quite sensitive to the "words of few mouths" phenomenon. Under "words of few mouths" scenarios not only the average performance degrades, but the performances of SVR and ANN also suffer severely from large stochastic variations.

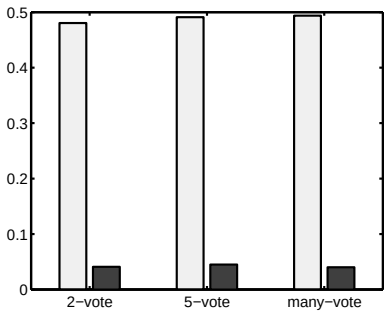


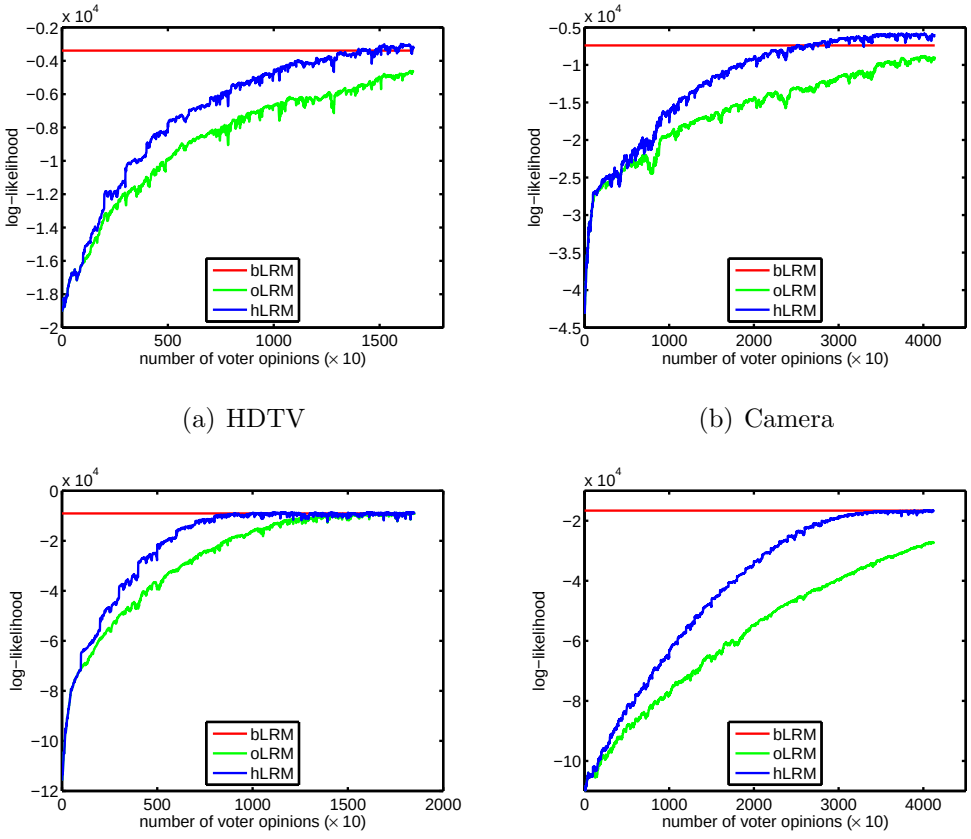
Figure 5.4: Comparison of the performances of LRM 2-vote, 5-vote and many-vote data. Empty bar: average of correlation values; solid bar: standard deviation of correlation values.

To further investigate and stress-test the robustness of the LRM algorithm, we also performed few-vote experiments with $k = 2$. Figure 5.4 compiles the performance results of LRM in many-vote experiments and few-vote experiments with $k = 5$ and $k = 2$ using the camera dataset. It can be seen from the figure that in fact even with only two votes on each review, the algorithm has the same average performance and

performance variation as the case using the many-vote dataset!

Finally, we would like to remark that the proposed batch algorithm is the most computationally efficient among the three algorithms. Compared with ANN, LRM runs slightly faster, and compared with SVR, LRM is much faster.

5.3.5 Online Algorithm Performance



(a) HDTV (b) Camera

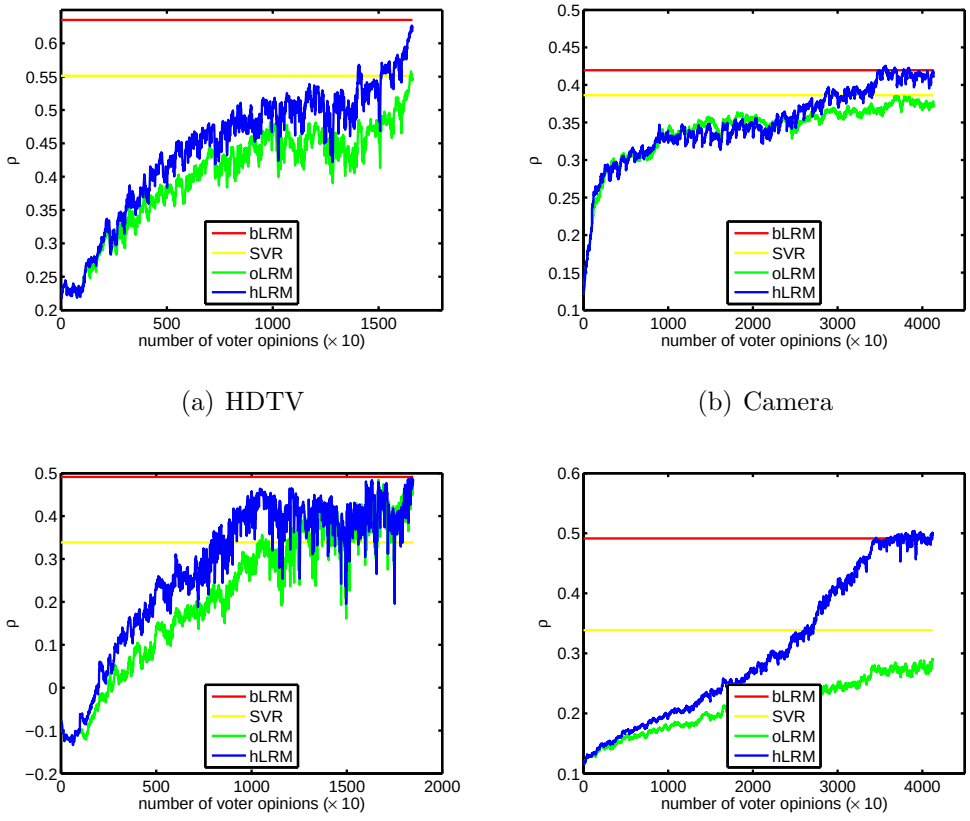
(c) Laptop (d) Book

Figure 5.5: The goodness-of-fit of batch and online algorithm.

To demonstrate the performance of oLRM and hLRM, we perform the following steps. At the beginning the experiments have no knowledge of consumer-generated reviews and voter opinions. oLRM operates on every 10 voter opinions arrival. We

randomly choose 10 voter opinions into the system and the performances of algorithms are evaluated after each new data arrivals. After the incoming data's arrival, the parameters of oLRM are updated and the performance on the testing set is evaluated. We also investigate the performance of bLRM and SVR with all the voter opinions available and the results are a constant when compared with the dynamic results from oLRM and hLRM.

Goodness-of-Fit Performance



(a) HDTV (b) Camera

(c) Laptop (d) Book

Figure 5.6: Spearman's rank correlation coefficient of oLRM.

Figure 5.5 demonstrates the goodness-of-fit of oLRM and hLRM as voter opinions incrementally arrive. SVR is not in the same probabilistic framework as oLRM, hLRM

and bLRM and so it is not included in this part. It can be seen that the log-likelihood of the testing set continues to increase as new data arrives. After accumulating enough voter opinions, oLRM and hLRM can reach the similar level of goodness-of-fit as the batch algorithm.

Moreover, as shown in Figure 5.5, the likelihood of hLRM is always greater than the likelihood oLRM as the sequence of voter opinions become available for all four categories of consumer-generated reviews. At the end of the consumer-generated review stream, hLRM can perform the same level of goodness-of-fit as bLRM, but for some categories of data oLRM does not perform at the same level. This fact confirms that introducing a batch phase in oLRM can improve the convergence rate of the online algorithm.

Ranking Performance

We analyze the helpfulness rank correlation obtained from bLRM, oLRM, hLRM and SVR to verify the ranking performance. The Spearman's rank correlation coefficient sequences as the consumer-generated review stream incrementally becomes available is shown in Figure 5.6. We can observe that the rank performance of bLRM is better than SVR over all four categories of consumer-generated reviews. This confirms the advantages of our probabilistic formulation and of our helpfulness modeling approach.

Figure 5.6 also shows that the ranking performances of hLRM are better than oLRM in terms of rank correlation and reach the performance level of bLRM faster than oLRM. This is because the batch phase of the hLRM increases the prediction performance of the online logistic regression. It can also be observed that the log-likelihood of helpfulness is consistent with ranking performance.

Based on above experiments, we have observed improvements for the helpfulness prediction problem with both online and off-line approaches. We may conclude that our proposed logistic regression model can effectively determine the helpfulness of

user-generated reviews. In practice, hLRM can start by collecting a small amount of voter opinions and the parameters of the model can be initialized by the batch algorithm, so that the performance will be further improved. We also would like to remark that the proposed algorithms are more computationally efficient then other existing approaches.

5.4 Discussion and Conclusion

In general, probabilistic modeling based inference and learning algorithms are particular suitable for handling uncertainty, errors and missing information in the dataset. In this chapter we have proposed probabilistic approaches to formulate the helpfulness of consumer-generated reviews in order to show the versatility of our helpfulness prediction framework, especially for the online helpfulness modeling.

In particular, we have introduced a batch algorithm and extended it to an online algorithm under the probabilistic modeling approach. To the best of our knowledge, this is the first online model which is built to assess the helpfulness of consumer-generated reviews. We have executed an empirical study on the consumer-generated reviews of Amazon.com and the experimental results show that our model can effectively and efficiently predict the helpfulness of consumer-generated review both online and off-line. Furthermore, we have designed a practical hybrid algorithm to start working with little or no knowledge of consumer-generated reviews and voter opinions to refine the model as it goes along. This hybrid algorithm learns online helpfulness models as new reviews enter into the system and intermittently runs a batch helpfulness model on the aggregated documents off-line. Such a hybrid model is useful for real-time applications; empirical studies show improvement on both the convergence rate and ranking performance of this hybrid model. By applying this model more helpful consumer-generated reviews can be found thus providing them helpful information.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Today's online customers are impatient and demanding while striving to ensure the best purchasing decisions. Meanwhile, they are not willing to spend much time and effort in comparing their potential products. The availability of consumer-generated reviews provides an immense amount of information that can assist consumers in making purchasing decisions. Consumers are willing to read others consumers' experiences and opinions; however, there is no formal format to sort through all the consumer-generated review contents available on the Internet. This collection of reviews is free-styled and consists of unstructured text and information. This fact renders consumers unable to find useful contents easily and thus they must spend more time surfing for helpful opinions and experiences.

The helpful consumer-generated reviews recommendation approaches proposed in this work can surely assist consumers in making purchasing decisions. An online store providing a review filtering system will certainly help potential consumers to find quality reviews and reduce their purchase decision making time. An online community that incorporates this review recommendation system can significantly increase their

usability and attract more users. One important and interesting extension of our model would be to integrate consumer-generated reviews from different sources or online communities. This would provide consumers with additional opportunities to benefit from other people's experience.

In this thesis, we have proposed a simple and efficient entropy-based approach to predict the helpfulness of consumer-generated reviews. This approach is suitable for applications which require the recommendation engine to be rapidly developed based on fully-voted reviews.

In addition to this, we have uncovered a widely existing phenomenon which we call "words of few mouths". This phenomenon presents additional challenges for developing machine-learning algorithms in recommender systems since the very few user opinions, if treated improperly, are either un-utilized (leading to lack of resources for learning), or are an additional source of "noise" in the training of the algorithms. The main philosophy advocated in this work is the use of probabilistic approaches to tackle such challenges where "words of few mouths" are treated as a sparse sampling of some distribution. Through developing a probabilistic learning algorithm for the review helpfulness prediction and comparing it rigorously against other machine-learning algorithms, we demonstrate the power of probabilistic methods in the presence of "words of few mouths".

Furthermore, we have proposed a logistic regression based-approach to learn helpfulness prediction model in real-time. We implement this model by a batch and an online algorithm. We compare the performance of the logistic regression-based batch algorithm (LRM) with two other popular machine learning algorithms - Support Vector Regression (SVR) and Artificial Neural Network (ANN). The experimental results show that LRM outperforms SVR and ANN. We also extend this model to an online algorithm to incrementally update the parameters of the helpfulness model from the incoming consumer-generated reviews. Finally, a hybrid algorithm for real-time helpfulness prediction systems is presented which learns online helpfulness models as re-

views are submitted and intermittently executes a batch algorithm on the aggregated reviews. The online component can ensure that the system is building a helpfulness model in real-time and the batch component can improve the model thoroughly on all the collected reviews off-line.

Although this thesis primarily focuses on helpfulness prediction for consumer-generated reviews, the general methodology presented in this thesis is applicable to the development of algorithmic engines for other recommender systems utilizing scaling and voting from users. In general, probabilistic modeling based inferences and learning algorithms are particular suitable for handling uncertainty, errors and missing information in the data set. The superior performance and robustness of the presented algorithm in review helpfulness prediction is merely one demonstration of the power of probabilistic algorithms.

This thesis uses unigram bag-of-word models however we note that probabilistic modeling frameworks in fact allow the use of more sophisticated features and more complex probabilistic models to describe the dependency between the compontents of our framework. The last two chapters in this thesis have shown the versatility and compatibility of the proposed framework.

6.2 Future Work

The helpfulness modeling of consumer-generated reviews is a relatively new research area in the domain of text mining and knowledge discovery. A number of techniques can be studied to further improve the effectiveness of our framework. In this section we discuss some possible future research directions.

6.2.1 Feature Set and Feature Selection

Feature selection is a step that reduces the set of features utilized in the learning process. Until now we have provided an algorithm that utilizes terms as the feature set, but we have not examined how our framework used with other feature sets can improve the existing method. Bag-of-N-grams model, where we create a vector that represents a document based on pairs, triples or even N-tuples of words, has been widely used in the natural language processing approaches to represent documents and information. We plan to investigate whether bigrams and n-grams will enhance the helpfulness discovering ability of our model as they have been employed by previous works [Tan et al., 2002; Mason et al., 2009] to improve the categorization performance of web pages. We have done a preliminary empirical study on comparing the performance of different commonly-used machine learning algorithms by using n-gram features and the results are listed in Appendix A. Constructing a domain specific dictionary for each learning domain might be used to reduce the feature dimensions and increase the performance of our algorithms.

Another direction would be to incorporate into our algorithms other factors that may affect the helpfulness of a review, such as semantic features, therefore improving the precision of recommendations.. In this thesis we assumed that all consumers have similar preferences for online reviews and did not consider individual differences such as personality, age, gender, education, interests and other aspects. Consequently, we would like to investigate the issue of personalization and consider the similarity and dissimilarity of consumers in our model in order to generate personalized recommendations of helpful reviews for different consumers.

6.2.2 Future Research on Helpfulness Modeling

We note that more complex probabilistic models are required to describe the dependency between different features and the dependency between features of “items”,

users and user opinions. For this purpose, there are several well-known graphical languages, such as Bayesian networks [Neapolitan, 2003] and Markov random fields [Kindermann, 1980], which one may use to build complex probabilistic models. In addition, there are families of well-established algorithms in these graphical modelling frameworks that allow for a principled approach to developing learning algorithms.

Helpfulness Modeling

A proper choice of distributions will enhance the ability of inferring helpfulness and so we have proposed a Gaussian-based approach to build the helpfulness model. Sometimes, only one distribution cannot be supplemented to give a deterministic theory. In order to investigate whether other distributions are more proper to be utilized, a richer class of density functions for modeling the helpfulness of a review document is needed. We plan to make use of other distributions or other combinations of distributions, like the Gaussian Mixture Model (GMM) [Duda and Hart, 1973], to simulate the relationship between the features and voter opinions in an attempt to improve the performance and accuracy of the helpfulness distribution prediction. Appendix B is an example under our proposed probabilistic framework and shows a helpfulness inference process by GMM.

Evaluation Metrics

As we discussed before, the proper selection of distribution is very important for inferring helpfulness. The goodness-to-fit becomes a strong indicator for choosing one distribution or combination of distributions over others.

We have proposed the log-likelihood based methodology to demonstrate the goodness-of-fit difference of different algorithms. In information theory the Kullback Leibler divergence (KL-Divergence) [Kullback and Leibler, 1951] is a measure of the dissimilarity between two determined probability distributions. Akaike's Infor-

mation Criterion (AIC) [Bozdogan, 2000] is another popular selection model criterion based on KL-Divergence. We would like to investigate the possibility of using these two model selection metrologies to improve our likelihood ratio based model selection criterion.

Alternative metrics for measuring review helpfulness will also be explored, including how our ranking metric can be improved to decide which helpfulness ranking metric is optimal for our helpfulness discovering problem. Even though we currently derive our ranking metrics by calculating a Bayesian estimation of the probability of a review document being a helpful one at a predefined degree, this scale needs further investigation. For instance, we can incorporate different helpfulness thresholds, like the mean helpfulness value in the training set, in order to discover the performance of a review recommendation algorithm.

In our probabilistic framework we evaluate the ranking performance by computing the Spearman's Ranking Correlation between algorithms. Other popular ranking performance evaluation measures including MAP (Mean Average Precision) [Turpin and Scholer, 2006] and NDCG (Normalized Discounted Cumulative Gain) [Järvelin and Kekäläinen, 2002] are more frequently used as ranking performance evaluation metric for information retrieval systems. These two retrieval measures might be examined in future studies to compare the ranking performance with other algorithms. We would like to do more thorough comparisons of our models to other state-of-the art text mining algorithms.

6.2.3 Possible Extensions

Applying social network information to the electronic commerce and social media domains definitely will discover more consumer behavior patterns and make services more user-friendly. Therefore, our approaches may be extended by incorporating the consumer social network information for generating more accurate recommendations.

One possible extension of our models is to introduce user clustering approaches which map users with common interest and similar behavioral patterns. Users' information may be used to split users of common interest into the groups to enhance their experience in interacting with other users further engaging the users, to recommend to each user cluster the items that they may be interested in consuming and to recommend to each user's potential friends and groups he/she may be interested in.

In addition, to incorporate item clustering models into our models, in conjunction with user clustering, will further improve the helpfulness discovery models' ability to intelligently discover potential items that a consumer may be interested in and recommend to the potential consumers.

Appendix A

A Preliminary Research on Incorporating N-Gram Features

N-gram is a subsequence of n words of a given document. As an example for n=2, the review document t,"it was considered the top of the line and it was expensive too", can be represented by a set of bigram features {it was, was considered, consider the, the top, top of, of the, the line, line and, and it, it was, was expensive, expensive too}.

We implement a three-layer error back-propagation (BP) ANN in this study. The number of neurons in its hidden layer is chosen to be 10. Each node utilizes sigmoid transfer function. The output node makes use of log-sigmoid transfer function. The training progress is set to be stopped after 1000 iterations of learning, or when present error value less than 0.001 is reached. We use the LibSVM [Chang and Lin, 2001] tool to perform the Support Vector Regression (SVR) learning algorithm. The parameters of the SVR, C and g, were chosen by applying a 10-fold cross-validation and a grid search on a logarithmic scale. We also apply the multiple linear regression (MLR) algorithm to find the mathematical relationship between features. The philosophy of MLR is to create a straight line that best approximates all input data points.

Table A.1: Spearman's rank correlation of the helpfulness prediction using different feature sets for different categories of review documents (10-fold cross-validation).

	SVR	ANN	MLR
GPS Reviews			
Unigram	0.4503	0.4238	0.4473
Bigram	0.3271	0.2243	0.4173
Trigram	0.3855	0.2567	0.2247
Unigram+Bigram	0.4631	0.3652	0.4545
Unigram+Bigram+Trigram	0.4632	0.3913	0.4541
LCD TV Reviews			
Unigram	0.5763	0.5083	0.5265
Bigram	0.3929	0.3368	0.4675
Trigram	0.0469	0.2498	0.2120
Unigram+Bigram	0.5627	0.5414	0.5450
Unigram+Bigram+Trigram	0.5651	0.4828	0.5476
Digital Camera Reviews			
Unigram	0.4533	0.4363	0.4200
Bigram	0.3150	0.2426	0.2810
Trigram	0.2420	0.2911	0.1936
Unigram+Bigram	0.4491	0.4446	0.4378
Unigram+Bigram+Trigram	0.4441	0.4420	0.4389

Table A.1 shows that the ranking performance of support vector regression (SVR), artificial neural network (ANN) and multiple linear regression (MLR) by applying to three different categories of online reviews. The 10-fold cross-validation process was used for the evaluation experiments. In this process, we use unigram, bigram, trigram, and the combination of these n-grams to represent the online product reviews. The results show that unigram features representation outperforms the bigram and trigram of features across all these machine learning algorithms. The reason why unigram outperforms bigram and trigram is that there are no sufficient data for training the model in our corpus and the sparsity of the resulted document-term matrix is very high.

Appendix B

Model the Dependency of Voter Opinions and Review Features by the Gaussian Mixture Model

Gaussian Mixture Model (GMM) assumes that the data follow a Gaussian Mixture distribution. In our case, we suppose that review features is generated for several Gaussian distributions. Each Gaussian distribution will be assigned a weight. Under our proposed helpfulness prediction framework proposed in Chapter 4, we model the dependency of features **F** on voter opinion **V** by the Gaussian Mixture Model. Figure B.1 is a Bayesian network [Neapolitan, 2003] representation of this model. More specifically, we define:

$$p(f_i|v_j = 1) = \sum_{i}^{k_1} \pi_i \mathcal{N}(f_i; \hat{\mu}_i, \Sigma_i), \tag{B.0.1}$$

and

$$p(f_i|v_j = 0) = \sum_{i}^{k_2} \pi_j \mathcal{N}(f_i; \hat{\mu}_i, \Sigma_i) \tag{B.0.2}$$

where k_1 and k_2 are the number of distributions, μ_i and μ_j are the means of distribution and Σ_i and Σ_j are the variances.

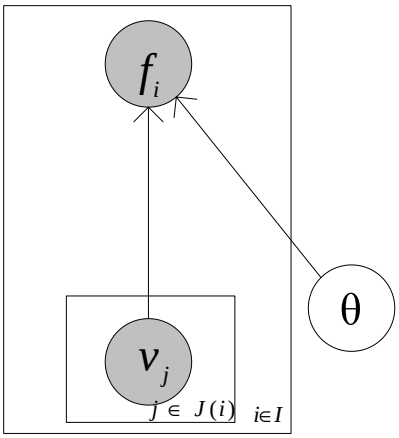


Figure B.1: Graphical model representation of the Gaussian Mixture Model

As we assume each review documents are independent, the objective function of the helpfulness prediction problem becomes to find a distribution which maximizes the observed data. It follows that:

$$p(f_I|v_J = 1) = \prod_{i \in I} \sum_{i}^{k_1} \pi_i \mathcal{N}(f_i; \hat{\mu}_i, \Sigma_i). \tag{B.0.3}$$

From Bayes' Theorem, the posterior density of voters' positive and negative opinion given review features is:

$$p(v = 1|f) = \frac{p(v = 1, f)}{p(f)} \propto p(f|v = 1)p(v = 1), \tag{B.0.4}$$

$$p(v = 0|f) = \frac{p(v = 0, f)}{p(f)} \propto p(f|v = 0)p(v = 0), \tag{B.0.5}$$

and

$$p(v_J|f_I, \theta) \propto p(v_J, f_I|\theta) = p(f_I|v_J; \theta)p(v_J). \tag{B.0.6}$$

EM Algorithm

We use Expectation Maximization algorithm to find parameters which maximize B.0.1 and B.0.2. Expectation Step:

$$E(z_{i,j}) = \frac{\pi_i^t p(f_i|\mu_i^t, \Sigma_i^t)}{\sum_{p=1}^{M} \pi_p p(f_p|\mu_p^t, \Sigma_p^t)} \tag{B.0.7}$$

Maximization Step:

$$\pi_i^{t+1} = \frac{1}{N} \sum_{j=1}^{N} E(z_{ij}) \tag{B.0.8}$$

$$\mu_i^{t+1} = \frac{1}{N\pi_i^{t+1}} \sum_{j=1}^{N} E(z_{ij}) f_j \tag{B.0.9}$$

$$\Sigma_i^{t+1} = \frac{1}{N\pi_i^{t+1}} \sum_{j=1}^{N} E(z_{ij}) \{(f_j - \mu_i^{t+1})(f_i - \mu_i^{t+1})^T\} \tag{B.0.10}$$

Bibliography

G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:734–749, 2005.

L. AlSumait, D. Barbará, and C. Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *ICDM '08: Proceedings of the 8th IEEE International Conference on Data Mining*, December 15-19, 2008, Pisa, Italy. IEEE Computer Society.

A. Anagnostopoulos, A. Z. Broder, and K. Punera. Effective and efficient classification on a search-engine model. *Knowledge and Information System*, 16(2):129–154, 2008.

D. Andrzejewski and X. Zhu. Latent dirichlet allocation with topic-in-set knowledge. In *SemiSupLearn '09: Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 43–48, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

M. Balabanovic. An adaptive web page recommendation service. In *Agents'97: Proceedings of the First International Conference on Autonomous Agents*, pages 378–385, New York, 1997. ACM.

I. Bíró, D. Siklósi, J. Szabó, and A. A. Benczúr. Linked latent dirichlet allocation in web spam filtering. In *AIRWeb '09: Proceedings of the 5th International Workshop*

on Adversarial Information Retrieval on the Web, pages 37–40, New York, NY, USA, 2009. ACM.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1 edition, January 1996.

D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 113–120, New York, NY, USA, 2006. ACM.

D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

H. Bozdogan. Akaike's information criterion and recent developments in information complexity. *Journal of Mathematical Psychology*, 44(1):62–91, March 2000.

P. B. Brazdil and C. Soares. A comparison of ranking methods for classification algorithm selection. In *ECML '00: Proceedings of the 11th European Conference on Machine Learning* , pages 63–74, Barcelona, Catalonia, Spain, 2000. Springer-Verlag.

J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann, 1998.

W. L. Buntine. Graphical models for discovering knowledge. *Advances in Knowledge Discovery and Data Mining*, pages 59–83. MIT Press, 1995.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.

D. Cai, Q. Mei, J. Han, and C. Zhai. Modeling hidden topics on document manifold. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 911–920, New York, NY, USA, 2008. ACM.

S. Castagnos, A. Brun, and A. Boyer. Probabilistic association rules for item-based recommender systems. In *STAIRS '08: Proceedings of the 4th Starting AI Researchers' Symposium*, pages 36–46, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.

C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines*, 2001. URL `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

K.-W. Cheung and L. F. Tian. Learning user similarity and rating style for collaborative recommendation. *Information Retrieval*, 7:395–410, 2004.

C. Danescu-Niculescu-Mizil, G. Kossinets, J. M. Kleinberg, and L. Lee. How opinions are received by online communities: a case study on amazon.com helpfulness votes. In *WWW '09: Proceedings of the 18th international conference on World Wide Web*, pages 141–150, Madrid, Spain, April 20-24, 2009. ACM.

S. R. Das and M. Y. Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388, 2007.

K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 519–528, New York, NY, USA, 2003. ACM.

G. N. Demir, A. S. Uyar, and S. G. Ögüdücü. Graph-based sequence clustering through multiobjective evolutionary algorithms for web recommender systems. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1943–1950, New York, NY, USA, 2007. ACM.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977a.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977b.

W. Duan, B. Gu, and A. B. Whinston. Do online reviews matter? – an empirical investigation of panel data. *Decision Support Systems*, 45(4):1007 – 1016, 2008.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Willey & Sons, New Yotk, 1973.

J. Euzenat. Semantic precision and recall for ontology alignment evaluation. In *IJCAI '07: Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 348–353, Hyderabad, India, 2007.

D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

V. Hatzivassiloglou and K. R. McKeown. Predicting the semantic orientation of adjectives. In *EACL '97: Proceedings of the 8th conference on European chapter of the Association for Computational Linguistics*, pages 174–181, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, New York, NY, USA, 1999a. ACM.

T. Hofmann. Probabilistic latent semantic analysis. In *UAI99: In Proceedings of Uncertainty in Artificial Intelligence*, pages 289–296, Stockholm, Sweden, 1999. Morgan Kaufmann.

D. W. Hosmer and S. Lemeshow. *Applied logistic regression (Wiley Series in probability and statistics)*. Wiley-Interscience Publication, 2 edition, 2000.

M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM.

Y. Huang and L. Bian. A bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the internet. *Expert Systems with Applications: An International Journal*, 36(1):933–943, 2009.

Z. Huang, D. Zeng, and H. Chen. A comparison of collaborative-filtering recommendation algorithms for e-commerce. *IEEE Intelligent Systems*, 22:68–78, 2007.

K. Inui, S. Abe, K. Hara, H. Morita, C. Sao, M. Eguchi, A. Sumida, K. Murakami, and S. Matsuyoshi. Experience mining: Building a large-scale database of personal experiences and opinions from web documents. In *WI-IAT '08: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 314–321, Washington, DC, USA, 2008. IEEE Computer Society.

T. Iwata, T. Yamada, Y. Sakurai, and N. Ueda. Online multiscale dynamic topic models. In *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 663–672, New York, NY, USA, 2010. ACM.

K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information System*, 20(4):422–446, 2002.

T. Jebara. *Machine Learning : Discriminative and Generative (The Kluwer International Series in Engineering and Computer Science)*. Springer, December 2003.

T. Jebara and M. Meila. Machine learning: Discriminative and generative. *The Mathematical Intelligencer*, 28:67–69, 2006.

I. T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, October 2002.

M. Jordan. Why the logistic function? a tutorial discussion on probabilities and neural networks. Technical report, Massachusetts Institute of Technology, 1995.

M. Jordan. *Learning in graphical models*. MIT Press, Cambridge, MA, USA, 1999.

M. Karimzadehgan, C. Zhai, and G. Belford. Multi-aspect expertise matching for review assignment. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1113–1122, New York, NY, USA, 2008. ACM.

S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti. Automatically assessing review helpfulness. In *EMNLP 06': Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 423–430, Sydney, Australia, July 2006. Association for Computational Linguistics.

R. Kindermann. *Markov Random Fields and Their Applications (Contemporary Mathematics ; V. 1)*. American Mathematical Society.

R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 61–68, New York, NY, USA, 2009. ACM.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:49–86, 1951.

C. Lee and G. G. Lee. Information gain and divergence-based feature selection for machine learning-based text categorization. *Information Processing and Management*, 42(1):155 – 165, 2006.

J. Lee, D.-H. Park, and I. Han. The effect of negative online consumer reviews on product attitude: An information processing view. *Electronic Commerce Research and Applications*, 7(3):341 – 352, 2008.

W. Lin, S. A. Alvarez, and C. Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6(1):83–105, 2002.

Y. Liu, X. Huang, A. An, and X. Yu. Modeling and predicting the helpfulness of online reviews. In *ICDM '08: Proceedings of the 8th IEEE International Conference on Data Mining*, pages 443–452, Pisa, Italy, 2008. IEEE Computer Society.

Y. Lu and C. Zhai. Opinion integration through semi-supervised topic modeling. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 121–130, New York, NY, USA, 2008. ACM.

Y. Lu, C. Zhai, and N. Sundaresan. Rated aspect summarization of short comments. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 131–140, New York, NY, USA, 2009. ACM.

J. E. Mason, M. Shepherd, and J. Duffy. Classifying web pages by genre: An n-gram approach. In *WI-IAT '09: Proceedings of 2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* , pages 458–465, Milan, Italy, 2009. IEEE Computer Society.

Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 171–180, New York, NY, USA, 2007. ACM.

R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *ACM DL '99: Proceedings of 15th ACM Conference on Digital Libraries*, pages 195–204, Berkeley, CA, USA, 1999. ACM.

N. Hu, L. Liu and J. J. Zhang. Do online reviews affect product sales? the role of reviewer characteristics and temporal effects. *Information Technology and Management*, 2008.

R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, April 2003.

H. S. Nielsen-Online. 81 percent of online holiday shoppers read online customer reviews, according to nielsen online, 12 2008. URL `www.nielsen-online.com/pr/pr_081218.pdf`.

N. J. Nnadi. Applying relevant set correlation clustering to multi-criteria recommender systems. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 401–404, New York, NY, USA, 2009. ACM.

A. Palmer. Web shoppers trust customer reviews more than friends. Online, 9 2009. URL `http://www.managesmarter.com/msg/search/article_display.jsp?vnu_content_id=1004011176`.

B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

D.-H. Park and S. Kim. The effects of consumer knowledge on message processing of electronic word-of-mouth via online consumer reviews. *Electronic Commerce Research and Applications*, 7(4):399 – 410, 2008.

D.-H. Park and J. Lee. ewom overload and its effect on consumer behavioral intention depending on consumer involvement. *Electronic Commerce Research and Applications*, 7(4):386 – 398, 2008.

D.-H. Park, J. Lee, and I. Han. The effect of on-line consumer reviews on consumer purchasing intention: The moderating role of involvement. *International Journal of Electronic Commerce*, 11(4):125–148, 2007.

J. F. Parra and S. Ruiz. Consideration sets in online shopping environments: the effects of search tool and information load. *Electronic Commerce Research and Applications*, 8(5):252–262, 2009.

I. Pollach. Electronic word of mouth: A genre analysis of product reviews on consumer opinion web sites. In *HICSS '06: Proceedings of the 39th Hawaii International Conference on Systems Science*, 3:51C, Kauai, HI, USA, 2006. IEEE Computer Society.

A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *UAI '01: Proceedings of the 17th conference on Uncertainty in artificial intelligence*, pages 437–444, 2001.

P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, New York, NY, USA, 1994. ACM.

B. B. Robert M. Schindler. *Online Consumer Psychology: Understanding and Influencing Consumer Behavior in the Virtual World*. Lawrence Erlbaum Associates, Inc., 2005.

M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '10: Proceedings of the 10th International World Wide Web Conference*, pages 285–295, Hong Kong, China, 2001. ACM.

C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

C.-M. Tan, Y.-F. Wang, and C.-D. Lee. The use of bigrams to enhance text categorization. *Information Processing and Management*, 38(4):529–546, 2002.

P. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics.

A. Turpin and F. Scholer. User performance versus precision measures for simple search tasks. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 11–18, New York, NY, USA, 2006. ACM.

I. E. Vermeulen and D. Seegers. Tried and tested: The impact of online hotel reviews on consumer consideration. *Tourism Management*, 30(1):123 – 127, 2009.

X. Wang, A. McCallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *ICDM '07: Proceedings of the 17th IEEE International Conference on Data Mining*, pages 697 –702, Omaha, Nebraska, USA, 2007. IEEE Computer Society.

M. Weimer, I Gurevych. Predicting the perceived quality of web forum posts. In *RANLP 07: Proceedings of the Conference on Recent Advances in Natural Language*

Processing, pages 643-648, Borovets, Bulgaria, 2007, Association for Computational Linguistics.

M. Weimer, I. Gurevych, and M. Mühlhäuser. Automatically assessing the post quality in online discussions on software. In *ACL '07: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 125–128, Prague, Czech Republic, 2007. Association for Computational Linguistics.

T.-L. Wong and W. Lam. Learning to extract and summarize hot item features from multiple auction web sites. *Knowledge and Information System*, 14(2):143–160, 2008.

X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. F. M. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and Information System*, 14 (1):1–37, 2008.

Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML '97: Proceedings of the 14th International Conference on Machine Learning*, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann.

H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP '03: Proceedings of the 2003 Conference on Empirical methods in natural language processing*, pages 129–136, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

B. Yuwono and D. L. Lee. Search and ranking algorithms for locating resources on the world wide web. In *ICDE '96: Proceedings of the Twelfth International Conference*

on Data Engineering, pages 164–171, Washington, DC, USA, 1996. IEEE Computer Society.

Z. Zhang and B. Varadarajan. Utility scoring of product reviews. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 51–57, New York, NY, USA, 2006. ACM.

Z. Zhang, Q. Ye, R. Law, and Y. Li. The impact of e-word-of-mouth on the online popularity of restaurants: A comparison of consumer reviews and editor reviews. *International Journal of Hospitality Management*, 29(4):694 – 700, 2010.

D. Zhou and Y. He. A hybrid generative/discriminative framework to train a semantic parser from an un-annotated corpus. In *COLING '08: Proceddings of the 22nd International Conference on Computational Linguistics*, pages 1113–1120, Manchester, UK, 2008. Association for Computational Linguistics.

F. Zhu and X. M. Zhang. Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics. *Journal of Marketing*, 74(2):133–148, 2010.

L. Zhuang, F. Jing, and X.-Y. Zhu. Movie review mining and summarization. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50, New York, NY, USA, 2006. ACM.