World Wide Web (2012) 15:117–138
DOI 10.1007/s11280-011-0127-3

Opinion helpfulness prediction in the presence of "words of few mouths"

Richong Zhang · Thomas Tran · Yongyi Mao

Received: 8 September 2010 / Revised: 5 January 2011 /
Accepted: 23 March 2011 / Published online: 14 April 2011

Abstract This paper identifies a widely existing phenomenon in social media content, which we call the "words of few mouths" phenomenon. This phenomenon challenges the development of recommender systems based on users' online opinions by presenting additional sources of uncertainty. In the context of predicting the "helpfulness" of a review document based on users' online votes on other reviews (where a user's vote on a review is either `HELPFUL` or `UNHELPFUL`), the "words of few mouths" phenomenon corresponds to the case where a large fraction of the reviews are each voted only by very few users. Focusing on the "review helpfulness prediction" problem, we illustrate the challenges associated with the "words of few mouths" phenomenon in the training of a review helpfulness predictor. We advocate probabilistic approaches for recommender system development in the presence of "words of few mouths". More concretely, we propose a probabilistic metric as the training target for conventional machine learning based predictors. Our empirical study using Support Vector Regression (SVR) augmented with the proposed probability metric demonstrates advantages of incorporating probabilistic methods in the training of the predictors. In addition to this "partially probabilistic" approach, we also develop a logistic regression based probabilistic model and correspondingly a learning algorithm for review helpfulness prediction. We demonstrate experimentally the superior performance of the logistic regression method over SVR, the prior art in review helpfulness prediction.

Keywords social media · recommender system · online review · helpfulness

R. Zhang (✉) · T. Tran · Y. Mao
School of Information Technology and Engineering, University of Ottawa,
800 King Edward Avenue, Ottawa, ON, K1N6N5, Canada
e-mail: rzhan025@site.uottawa.ca

T. Tran
e-mail: ttran@site.uottawa.ca

Y. Mao
e-mail: yymao@site.uottawa.ca

1 Introduction

"Electronic word of mouths" [26], or EWOM, on the social media web sites, may widely refer to information, opinions and user inputs of various kinds, which are provided independently by the Internet users. In the modern age of the Internet, EWOM, or user-generated content, has become a central component and attractive feature of social media sites. The fact that there is an enormous amount of information contained in EWOM makes building recommender systems based on EWOM very appealing. Indeed, significant research efforts have been spent over the past years along this direction (see, e.g., [1], for a comprehensive review of the area).

In a grand scope, this paper deals with the problem of developing recommender systems for social media web sites from the EWOM therein. In general, such a problem may be abstracted in terms of a collection of *"widgets"*, a collection of *users*, and some users' *feedbacks* on a subset of the "widgets". Here, a "widget" can refer to a movie, a video clip, a product, a blog, an article, etc; the feedback of a user on a widget could be in text form (such as an review article), numerical form (such as a product rating), categorical form (such as tags), binary form (such as `LIKE`/`DISLIKE`) etc. The objectives of developing the recommender system in this context may include deciding which widgets are to be recommended to a particular user or to a typical user, deciding what level of recommendation should be given, etc.

Narrowing down to a specific case of recommender systems, in this paper we will primarily consider developing algorithms for predicting the helpfulness of user opinions based on the votes from other users and more specifically, we focus on developing "review helpfulness predictors". In this context, each "widget" is a review (of a movie, a video, a commercial product, etc.) written by a user, and the user feedbacks are in binary forms, namely, that each user having read a review may vote the review `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. Despite the simplicity of its formulation, a good opinion helpfulness predictor is an important component of a social media web site, since it allows the web site host to automatically discover user opinions that are of interest to other users.

User opinions and other users feedbacks on these opinions, have their unique characteristics in the context of recommender systems. Albeit the richness of their information content, the opinions provided by individual users are typically less formal, often biased, and have relatively low reliability and large quality variation. As such, retrieving information from such sources is typically a challenging task. In addition, this challenge is often amplified by a phenomenon which we call "words of few mouths". Here, "words of few mouths", a term we coin in this paper, refers to the phenomenon where there is a large fraction of "widgets" each only having received feedbacks from very few users. For example, Figure 1 plots the histogram of the

[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.

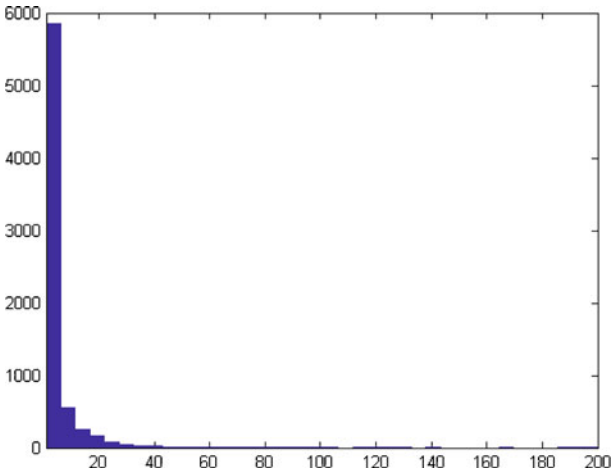


Figure 1 Histogram of the number of votes in the review document data set at Amazon.com. X-axis represents the number of votes that the review document has received. Y-axis represents the number of review documents receiving a certain number of votes specified on x-axis (0–5, 6–10, etc.)

number of votes on 9,955 reviews at Amazon.com, where there are more than 50% of the reviews each voted by no more than five users. The "words of few mouths" phenomenon of Amazon.com shown in this figure is not an isolated case. In fact, such a phenomenon widely exists in many E-commerce web sites. The underlying reason for this phenomenon is perhaps that except for a small number of "popular" ones, most "widgets" are "unpopular".

The challenges brought by the "words of few mouths" phenomenon in the development of 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 infer the dependency of positive vote fractions on review documents (see, e.g., [18, 27] and [21]). However, in the case where most reviews are each voted only by a few users, the positive vote fraction is a poor indicator of review helpfulness and the performance of the predictors trained this way necessarily degrade. We argue that the helpfulness of a review should only depend on the review document itself and on the *statistics* of the reader population, and unless there are a large number of votes for each review document, positive vote fraction 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, which depends on whether there is additional information available, the size of the data set, the heterogeneity of the "widgets" and that of "users", etc. A partial cure of the "words of few mouths" problem is to remove the "unpopular widgets" 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.

This paper advocates probabilistic approaches to developing recommender system from "words of few mouths", where we focus on the review helpfulness prediction problem. We first propose a probabilistic helpfulness metric for the conventional

machine learning based predictors. In addition and more significantly, we present a logistic regression based probabilistic model and 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. More specifically, we show that SVR with the proposed probabilistic helpfulness metric outperforms SVR with conventional helpfulness metric (positive vote fraction), but it is still inferior to the logistic regression method. This demonstrates the advantage of using probabilistic approaches in the presence of the "words of few mouths" phenomenon. A message we wish to convey is that partially augmenting standard learning methods with probabilistic ingredients may improve the performance of the learning machine, however, more significant performance gain is only attainable with by building the learning machine entirely from a probabilistic perspective.

To the best of our knowledge, the "words of few mouths" problem has not been noted in the literature of recommender systems. Although this paper primarily considers the problem of review helpfulness prediction, the presented probabilistic methodology is in general applicable for developing other recommender systems based on EWOM.

The remainder of this paper is organized as follows. Section 2 outlines the related works. Section 3 presents the basic probabilistic postulates underlying real-life data for review helpfulness prediction, and proposes two approaches for handling "words of few mouths". The first approach is by augmenting the conventional machine-learning predictors with a probabilistically derived helpfulness metric, and the second approach is a fully probabilistic approach based on logistic regression. Section 4 experimentally compares SVR with the conventional helpfulness metric (SVR-A), SVR augmented with the proposed probabilistic metric (SVR-B), and the logistic regression method (LRM), where we show that SVR-B outperforms SVR-A and LRM outperforms SVR-B. The paper ends with some discussion and brief conclusions in Section 5.

2 Related works

Recently, as the rapid development of social media, the large amount of web data overburdens the internet users. Researchers have proposed different approaches to ease the difficulty of users. Collaborative filtering recommender systems [8, 25], based on user ratings, generate recommendations about movies, music or news. Personalize website navigation system [7] helps users navigate through potentially interesting pages easily. Web content filtering systems [2, 9] assist users to retrieve related information from a large amount of data. The main intention of these studies is to generate recommendations to social media users based on different underlying approaches. However, users 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 would like to hear from other people who have used the products that they are now interested in purchasing.

Some work has been done in the area of review mining and summarizing. Zhuang et al. mined and summarized the movie reviews based on a multi-knowledge approach that included WordNet, statistical analysis and movie knowledge in [31]. Hu and Liu also summarized product reviews by mining opinion features [14].

Hatzivassiloglou and McKeown proposed a method to predict the semantic orientation of adjectives by a supervised learning algorithm in [10]. In [29], authors proposed a classification approach to separate sentences as positive or negative. In [24], authors classified movie reviews as positive or negative by several machine learning methods that were Naive Bayes, Maxium Entropy and Support Vector Machines and they also used different features like unigram, bigram, position and a combination of these features. The results shown that unigram presence feature was the most effective and the SVM performed the best for sentiment classification. These studies focus on sentiment classification and opinion mining for online reviews, however, only a few researches consider the usefulness of online review contents.

Kim et al. developed an SVM-based method to assess review helpfulness, where review lengths, unigrams and product ratings are taken as the discriminating features [18]. Weimer et al. introduced an algorithm to assess the quality of posts in web forums using a variety of features including surface, lexical and syntactic features, as well as some forum-specific features and certain similarity features in [28]. In a follow-up, Weimer and Iryna [27] extended the method into three data sets and found that the SVM classification performed the best.

In [21], authors presented a nonlinear regression model for the helpfulness prediction. Three groups of factors that might affect the value of helpfulness were analyzed and the model was build upon on these three groups of factors. The results of applying their model showed that the performance was better than the SVM regression model. In [30], authors incorporated a diverse set of features in an attempt to build a regression model to predict the utility of online product reviews. These works all heuristically choose parameters and no previous work has been done to build a generalized framework to model and evaluate the helpfulness of reviews.

The probabilistic modeling methodologies are capable of describing problems in the most concise way. Probabilistic Models have been employed into the text content analysis domain. Probabilistic LSA (LSA) [11, 12] and Latent Dirichlet Allocation (LDA) [4] are quickly becoming the most powerful probabilistic document modeling techniques and are accepted by a variety of text processing applications [3, 17, 20, 22]. These approaches provide greater insights and perspectives about real-world issues and the probabilistic inference gives a theoretical and mathematical description of the model. However, they are not suitable for analysis the helpfulness of social media content because they do not take the voters opinion into account. Most of the available algorithms for text analysis are related to the topic modeling. Also, there is no existing research using the probabilistic models to solve the helpfulness prediction problem. Our study focuses on proposing a concise probabilistic approach to model the helpfulness of user-generated content and to assist social media users to find the most helpful contents more efficiently.

3 Review helpfulness prediction

3.1 The problem

To formulate the review helpfulness prediction problem, 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 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 $\{J(i) : i \in I\}$.

3.2 Conventional approaches

The conventional approaches to the problem start by defining a helpfulness metric, then extracting the helpfulness metric from the data set (d_I, v_J) and using a machine-learning approach to infer the dependency of the helpfulness metric on the review document.

In all previous works, (see, for example, [18, 21, 27], etc), the helpfulness metric of a review is chosen as the "positive vote fraction" of the review observed from the data set. More precisely, the positive vote fraction α_i of review i is defined as the fraction of votes indexed by $J(i)$ that are equal to 1, namely,

$$\alpha_i := \frac{|J^+(i)|}{|J(i)|}, \tag{1}$$

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

Built on this choice of helpfulness metric, conventional approaches, including for example, such as SVR, start with extracting the positive vote fraction α_i for each review in d_I and attempts to infer the dependency of a 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. Methodologically, these approaches boil 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 not suitable for the case of "words of few mouths", since when a review is voted only by a small number voters, the extracted positive vote fraction is highly unreliable and suffers greatly from statistical irregularity.

In the remainder of the paper, we approach the helpfulness prediction problem from a probabilistic perspective.

3.3 Basic probabilistic postulates

We first present the basic probabilistic postulates that will be used throughout this paper.

Formally, Let $\mathcal{D}$ be the space of all reviews and $\mathcal{R}$ be the space of all functions mapping $\mathcal{D}$ to $\{0, 1\}$. Here each function $r \in \mathcal{R}$ is essentially a "voting function" characterizing a way to vote any document in $\mathcal{D}$. We consider that our data

$(d_I, v_J, \{J(i) : i \in I\})$ is the result of random sampling of the cartesian product space $\mathcal{D} \times \mathcal{R}$ based on the following postulates:

1. There is a distribution $p_{\mathbf{D}}$ on $\mathcal{D}$, i.i.d. sampling of which results in d_I.
2. For each $d \in \mathcal{D}$, there is a conditional distribution $p_{\mathbf{R}|d}$ on $\mathcal{R}$ and for each d_i, $i \in I$, the rating functions resulting from i.i.d. sampling of $p_{\mathbf{R}|d_i}$ gives rise to the set of votes $v_{J(i)}$ on review d_i via acting on d_i.

Here, and as well as will be followed throughout the paper, we have adopted the notations that random variables (and more generally random functions) are denoted by capitalized bold-font letters, a value that a random variable may take is denoted by the corresponding lower-cased letter, and any probability distribution is denoted by p with an appropriate subscript to indicate the concerned random variable(s). When it is clear from the context, we may drop the subscripts of p to lighten the notations.

The two postulates above present a generative interpretation of data $(d_I, v_J, \{J(i) : i \in I\})$. This allows us to characterize the *intrinsic helpfulness* h of a review document $d \in \mathcal{D}$ as *the probability that a random voting function* $\mathbf{R}$ *drawn from distribution* $p_{\mathbf{R}|d}$ *results in* $\mathbf{R}(d) = 1$, or *the probability that a random reader will vote review document* d HELPFUL.

We note that the joint distribution p_{DR} on the cartesian product space $\mathcal{D} \times \mathcal{R}$ induced by the above procedure also induces a conditional distribution $p_{\mathbf{V}|\mathbf{D}}$ on $\{0, 1\} \times \mathcal{D}$, where $\mathbf{V}$ takes values in $\{0, 1\}$ and $\mathbf{D}$ takes values in $\mathcal{D}$. This distribution is essentially the distribution of a random vote conditioned on a random document D, and the evaluation of this distribution at $\mathbf{V} = 1$ and $\mathbf{D} = d$, namely, $p_{\mathbf{V}|D}(1|d)$, equals the probability that $\mathbf{R}(d) = 1$, or the helpfulness of document d.

3.4 Augmenting conventional approaches with probabilistic helpfulness metric

Before we consider a completely probabilistic approach to the helpfulness prediction problem in the next subsection, we first consider augmenting the conventional approaches by incorporating the two probabilistic postulates thereby handling the uncertainty arising from "words of few mouths". Without altering the machine-learning approaches to the problem, here we propose to use a different helpfulness metric for machine learning. In particular, we will take a Bayesian approach to define a probabilistic helpfulness metric.

Returning to the two postulates, if $p_{\mathbf{R}|d}$ is given for every review document, then h, or $h(d)$, is a fixed number in [0, 1] for every d. However, since $p_{\mathbf{R}|d}$ is unknown to the learning machine, then in a Bayesian approach, we can in fact treat h as a random variable $\mathbf{H}$. Before observing any voting statistics for review document d, we assume that $\mathbf{H}$ is uniformly distributed in [0, 1] *a priori*, namely $p_{\mathbf{H}}(h) = 1$ for any $h \in [0, 1]$. Upon observing the voting statistics of document d, we may then use the Bayes Rule to determine the *a posteriori* probability $p_{\mathbf{H}_i|v_{J(i)}}$ as follows:

$$
\begin{aligned}
p_{\mathbf{H}_i|v_{J(i)}}(h) &= \frac{p_{\mathbf{V}|h}(1|h)^{|J^+(i)|}\, p_{\mathbf{V}|h}(0|h)^{|J^-(i)|}\, p_{\mathbf{H}}(h)}{\int_0^1 p_{\mathbf{V}|h}(1|h)^{|J^+(i)|}\, p_{\mathbf{V}|h}(0|h)^{|J^-(i)|}\, p_{\mathbf{H}}(h)dh} \\
&= \frac{h^{|J^+(i)|}(1-h)^{|J^-(i)|}}{\int_0^1 h^{|J^+(i)|}(1-h)^{|J^-(i)|}dh} \qquad (2)
\end{aligned}
$$

where according to Postulate 2, we have made use of the fact that given $\mathbf{H} = h$, $\mathbf{V}_{J(i)}$ is a set of i.i.d. Bernoulli random variables parameterized by h.

Based on this posterior of $\mathbf{H}_i$, we then use a different helpfulness metric, namely, $r_i := \Pr[\mathbf{H}_i > 0.5|v_{J(i)}]$, to replace the positive vote fraction for learning purposes. In particular, r_i may be computed by

$$r_i = \Pr[\mathbf{H}_i > 0.5|v_{J(i)}] = \int_{0.5}^{1} p_{\mathbf{H}_i|v_{J(i)}}(h)dh$$

This metric, instead of trying to measure how helpful is a review, measures how confidently one may claim a review to be helpful based on voting statistics if the intrinsic helpfulness value at 0.5 is used as the decision threshold. Although this choice of metric is to a degree heuristic, it may be sensibly justified as follows.

1. If one is to classify reviews into `HELPFUL` and `UNHELPFUL` classes based on the intrinsic helpfulness of the review, the most natural choice of decision threshold is 0.5.
2. In the "words of few mouth" regime, namely when the number of votes are small for most of the reviews, "how confidently one may claim a review to be helpful" intuitively mean the same thing as "how helpful a review is". In fact, the former perhaps serves better as the helpfulness metric.

Figure 2 shows the posterior distribution of $\mathbf{H}$ given the voting statistics and demonstrates the difference between positive vote fraction (α_i) and this probabilistic metric for different voting statistics. Seen from the figure, the advantage of using this metric is its ability to distinguish different voting statistics having same positive vote fraction are distinguished by this metric, particularly in the "words of few mouths" regime.

However, we must point out that the proposed metric has a significant drawback in when the data set primarily consists of review documents voted by a very large number of users. When the number of votes on a document is very large, there is a decrease of resolution in the metric's ability to indicate the helpfulness of a review. In that case, the metric tends to be concentrated either at 1 or at 0, corresponding to the positive vote fraction larger than 0.5 and smaller than 0.5 respectively. Fortunately for the scope of this study, this drawback is of little concern, since we only deal with data set in the "words of few mouths" regime.

As we will show later in this paper, the proposed metric does show performance advantages. We however believe that there are other metrics formulated probabilistically, which may also help to improve the performances of the machine learning-based predictors.

Instead of following the conventional machine-learning approaches, we now present a completely probabilistic approach for helpfulness prediction.

3.5 Review helpfulness prediction as probabilistic inference

Continuing from the two probabilistic postulates, the helpfulness prediction problem may be formulated as the following probabilistic inference problem: Given data $(d_I, v_J, \{J(i) : i \in I\})$ generated from the above procedure, determine the distribution $p_{\mathbf{V}|\mathbf{D}}$.

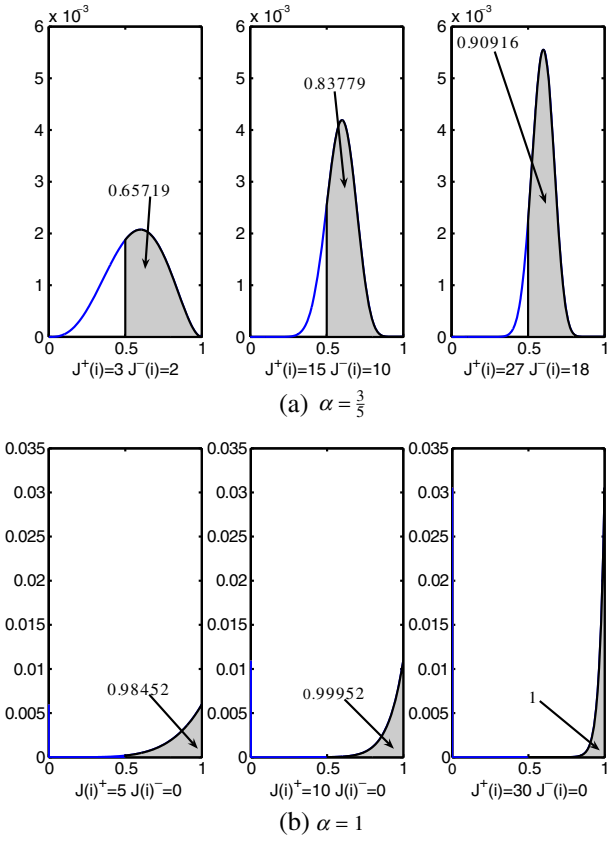


Figure 2 The probability density function $p_{\mathbf{H}_i|v_{J(i)}}$. **a** $\alpha_i = \frac{3}{5}$. **b** $\alpha_i = 1$

Here, 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 however also 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 the helpfulness prediction problems.

Although one may consider various options to adapt 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 model $p_{\mathbf{V}|\mathbf{D}}$, and then choose one of the candidates under which the (log)likelihood of observed 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

$$p^*_{\mathbf{V}|\mathbf{D}} := \underset{p_{\mathbf{V}|\mathbf{D}} \in \Theta_{\mathbf{V}|\mathbf{D}}}{\operatorname{argmax}} \log p_{V_J|D_I}(v_J|d_I) \tag{3}$$

Following the two postulates, we may re-write (3) as

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

$$= \operatorname*{argmax}_{p_{\mathbf{V}|\mathbf{D}} \in \Theta_{\mathbf{V}|\mathbf{D}}} \sum_{i \in I} \sum_{j \in J(i)} \log p_{\mathbf{V}|\mathbf{D}}(v_j|d_i) \tag{4}$$

As is common in many machine-learning problems, the huge dimensionality of space $\mathcal{D}$ makes solving problem (4) 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 (4) is then modified to finding

$$p^*_{\mathbf{V}|\mathbf{F}} = \operatorname*{argmax}_{p_{\mathbf{V}|\mathbf{F}} \in \Theta_{\mathbf{V}|\mathbf{F}}} \sum_{i \in I} \sum_{j \in J(i)} \log p_{\mathbf{V}|\mathbf{F}}(v_j| f_i), \tag{5}$$

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 paper, we present a prediction algorithm similar to the logistic regression algorithm [13] developed in classification literature.

3.6 Logistic Regression for Helpfulness Prediction

Central to solving the optimization problem specified in (5) 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 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* [16]. More precisely, we define

$$p_{\mathbf{V}|\mathbf{F}}(1| f) = \mu(\eta), \tag{6}$$

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.[2] We note that since $p_{\mathbf{V}|\mathbf{F}}(1|f) + p_{\mathbf{V}|\mathbf{F}}(0|f) = 1$, (6) completely defines model $\Theta_{\mathbf{V}|\mathbf{F}}$, namely,

$$\Theta_{\mathbf{V}|\mathbf{F}} := \left\{ p_{\mathbf{V}|\mathbf{F}} \text{ satisfying } p_{\mathbf{V}|\mathbf{F}}(1|f) = \frac{1}{1+e^{-\theta^T f}} : \theta \in \mathbb{R}^m \right\}, \tag{7}$$

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

We note that this model is valid since logistic function has range $(0, 1)$. In addition, 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 conational distribution of class label given feature is a logistic function. This fact together with the richness of the exponential family makes our choice of $\Theta_{\mathbf{V}|\mathbf{F}}$ a robust and general model, rather 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 3 is a Bayesian network [23] representation of the model.

Now using model $\Theta_{\mathbf{V}|\mathbf{F}}$ defined in (7), the optimization problem of (5) reduces to solving

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^m}{\operatorname{argmax}} \sum_{i \in I} \sum_{j \in J(i)} \left[v_j \log \mu(f_i) + (1 - v_j) \log (1 - \mu(f_i)) \right], \tag{8}$$

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

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

$$\begin{aligned}
\frac{dl}{d\theta} &= \sum_{i \in I} \sum_{j \in J(i)} \left[v_j \frac{1}{\mu(f_i)} \frac{d\mu(f_i)}{d\theta} + (1 - v_j) \frac{1}{1 - \mu(f_i)} \frac{d\mu(f_i)}{d\theta} \right] \\
&= \sum_{i \in I} \sum_{j \in J(i)} \left[v_j \frac{1}{\mu(f_i)} \mu(f_i)(1 - \mu(f_i)) f_i \right. \\
&\qquad \left. + (1 - v_j) \frac{1}{1 - \mu(f_i)} \mu(f_i)(1 - \mu(f_i)) f_i \right] \\
&= \sum_{i \in I} \sum_{j \in J(i)} v_j (1 - \mu(f_i)) f_i - \mu(f_i)(1 - v_j) f_i \\
&= \sum_{i \in I} \sum_{j \in J(i)} f_i (v_j - \mu(f_i))
\end{aligned} \tag{9}$$

[2] In this paper, all vectors are by default taken as column vectors.

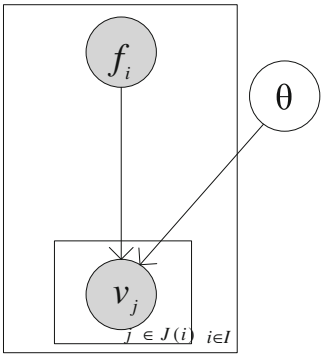


Figure 3 Graphical model representation

This allows a gradient ascent algorithm to optimize the objective function, in which value of the objective function can be step-by-step increased via updating the configuration of θ according to

$$\theta^{t+1} := \theta^t + \Delta \sum_{i \in I} \sum_{j \in J(i)} f_i(v_j - \mu(f_i)). \tag{10}$$

where Δ is a choice of step size.[3]

We articulate this algorithm as follows, where the superscript on θ indexes iteration number.

Algorithm 1 The gradient descent algorithm with dynamically changing step size. CC denotes the algorithm convergence condition and SSCC denotes the step size changing condition.

Input: f_I and v_J
Output: θ
Set CC;
Set SSCC;
$t := 0$;
set θ^t to a random configuration;
set Δ to a relatively large value;
$t := t + 1$;
while *CC is not satisfied* **do**
  Update θ^t by (10);
  if *SSCC is met* **then**
    Decrease Δ;
  end
  $\theta = \theta^t$
end

[3] As is well known in optimization literature, large step size allows the gradient ascent algorithm to hop away from the local optimums but suffers from large oscillation in the value of the objective function; small step size allows the algorithm to converge with small variation in the value of the objective function but suffers from higher chances of getting trapped at local optimums. In the implementation of our algorithm, in order to obtain the best compromise, we in fact gradually decrease the step size as the algorithm iterates.

We note that in this algorithm, we allow the step size Δ to decrease gradually. This is to 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 optima.

Finally, we remark that to be comparable with other machine-learning methods, we may view the helpfulness metric of the proposed logistic regression method as $p_{\mathbf{V}|\mathbf{F}}(1|f_i)$.

4 Experimental evaluation

To demonstrate the effectiveness of the proposed approach, we experimentally evaluate our logistic regression model (LRM) and compare it with the most commonly used machine learning method Support Vector Regression (SVR) [5]. This section presents our method of evaluation, experimental setups and results of comparison.

4.1 Method of evaluation

A difficulty associated with "words of few mouths" in evaluating the performances of different algorithms is the lack of benchmarks for those "unpopular widgets". In the context of helpfulness prediction, this difficulty translates to the question what to use as the true intrinsic helpfulness value of a review that is only voted by a few users.

To get around this difficulty, for a given real data set that will be used to evaluate the algorithms of interest, we remove the reviews that are voted by fewer than M users. We will refer to the resulting data set as the "many-vote" data set. It is apparent that when M is reasonably large, we may use the positive vote fraction to benchmark the helpfulness of the reviews in the many-vote data set. In this work, we choose $M = 15$.

We then construct a "few-vote" data set from the many-vote 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 uniformly, at random, from the set of $\{3, 4, 5, 6, 7\}$ 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 data set, each review is voted by k users.

We then partition the set of reviews into the set $\mathcal{N}$ of training reviews and the set $\mathcal{T}$ of testing reviews, where 2/3 of the reviews are training reviews and 1/3 are testing reviews. The partitioning is performed repeatedly using random subsampling, 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 $(\mathcal{N}, \mathcal{T})$'s are generated in our study.

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

Few-Vote Experiment For each real data set 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 data set. 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 data set.

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 data set is computed. The definition of ρ is given below.

$$\rho = 1 - \frac{6 \sum_{i \in \mathcal{T}} (x_i - y_i)^2}{|\mathcal{T}|(|\mathcal{T}|^2 - 1)}, \tag{11}$$

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 data set.

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

$$t := \frac{\bar{\rho}(A1) - \bar{\rho}(A2)}{S}, \tag{12}$$

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.

4.2 Experimental setup

As there is no standard customer review corpus available, we utilize the web services provided by Amazon.com to crawl the web site and obtain three data sets of review documents and vote information: HDTV data set, digital camera data set, and book data set. The HDTV many-vote data set contains 393 reviews and 12,058 votes; the camera many-vote data set contains 390 reviews and 12,003 votes; book many-vote data set contains 1,691 reviews and 48,057 votes.[4] In HDTV many-vote data set, the average number of votes on each review is 30.68 and 75% of reviews receive fewer

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

than 38 votes. In camera many-vote data set, the average number of voter opinions on each review is 31.03 and 75% of reviews have fewer than 36 votes. In book many-vote data set, the average number of voter opinions on each review is 28.41 and 75% of reviews have fewer than 30 reviews.

Since the objective of this work is not to develop a sophisticated language model but rather to study the "words of few mouths" problem, we use the simple "bag of words" language model to represent each review document, and the feature vector associated with each review document is simply taken as a binary {0, 1}-valued vector where each coordinate of the vector corresponds to a word, and "1" at any coordinate indicates the occurrence of the corresponding word in the review.

For each partition $(\mathcal{N}, \mathcal{T})$, prior to the training of the three algorithms, dimensionality reduction is performed on the feature vectors using the principal component analysis (PCA) [15]. We select the top 200 principal components in PCA, which accounts for 70% of the total variance.

Three algorithms are considered in our 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
- Logistic regression method, which will be denoted by LRM

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 review helpfulness in [18] and [30]. We implement the SVR algorithms using the LibSVM [6] toolkit. The parameters of the SVR, C and g, are chosen by applying a 10-fold cross validation and a grid search on a logarithmic scale.

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.

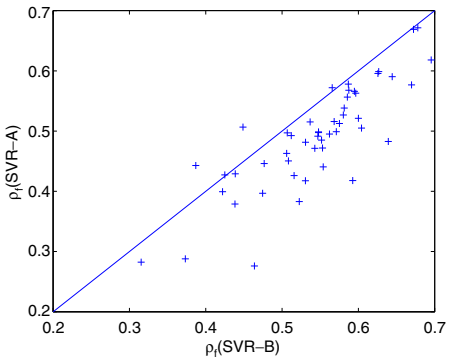
As mentioned earlier, depending on the type of experiments performed, the probabilistic helpfulness metric may be extracted either from the few-vote data set or from the many-vote data set.

4.3 Experimental results

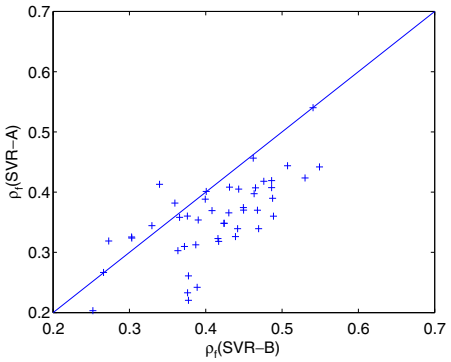
We analyze helpfulness rank correlation obtained from the three algorithms. Algorithms to verify the performance of different algorithms. First, we investigate the advantages of making use of the proposed probabilistic helpfulness metric as the learning target.

Figure 4 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 experiments. Each point in any plot corresponds to one partition $(\mathcal{N}, \mathcal{T})$. Here, for convenience, the subscript f and m of ρ refer to the few-vote and many-vote experiment respectively. It is visually apparent that the points in each of these plots primarily scatter above the $y = x$ diagonal line, suggesting that there is a significant performance advantage of SVR-B over SVR-A.

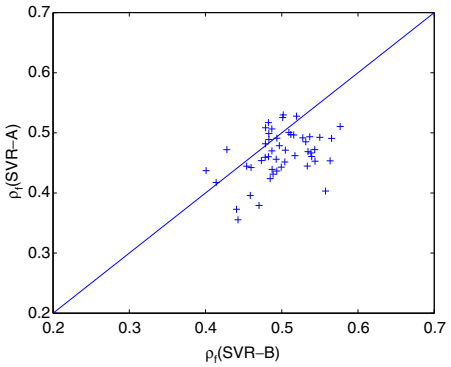
Figure 4 Comparison of helpfulness rank correlation using SVR-A and SVR-B on few-vote data sets



(a) SVR-A vs SVR-B (HDTV): $\bar{\rho}_f$(SVR-A)=0.490, $\bar{\rho}_f$(SVR-B)=0.544, p-value=4.40e-10



(b) SVR-A vs SVR-B (Camera): $\bar{\rho}_f$(SVR-A)=0.342, $\bar{\rho}_f$(SVR-B)=0.407, p-value=4.87e-10



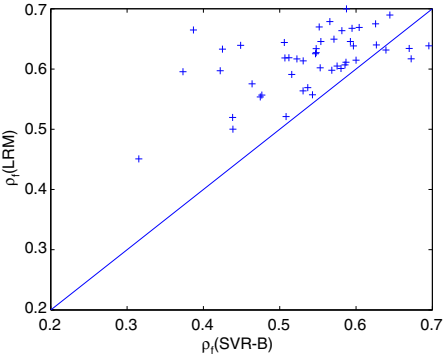
(c) SVR-A vs SVR-B (Book): $\bar{\rho}_f$(SVR-A)=0.462, $\bar{\rho}_f$(SVR-B)=0.4990, p-value=5.80e-7

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.

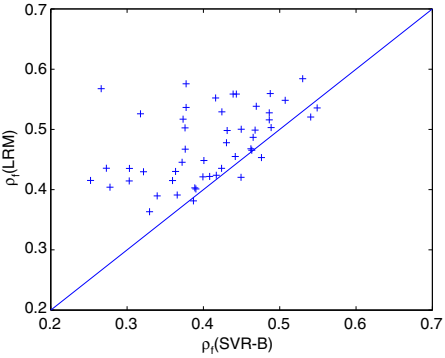
These results confirm that our proposed probabilistic metric is more suitable for the learning purpose when only a small number of votes available in the training set.

Since SVR-B outperforms SVR-A for few-vote data sets, we compare LRM with SVR-B for the few-vote case. Figure 5 shows a set of scatter plots that compare

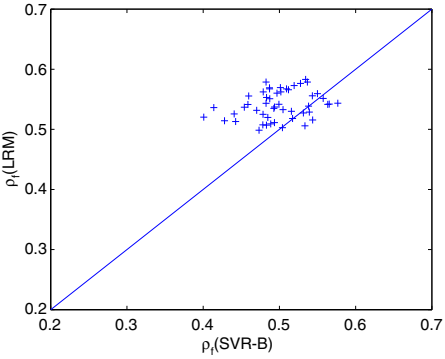
Figure 5 Comparison of helpfulness rank correlation between LRM and SVR-B using few-vote data sets



(a) LRM vs SVR-B (HDTV): $\bar{\rho}_f$(LRM)=0.618, $\bar{\rho}_f$(SVR-B)=0.544, p-value=2.47e-10



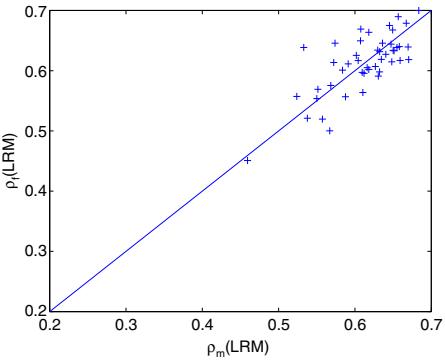
(b) LRM vs SVR-B (camera): $\bar{\rho}_f$(LRM)=0.475, $\bar{\rho}_f$(SVR-B)=0.406, p-value=5.89e-9



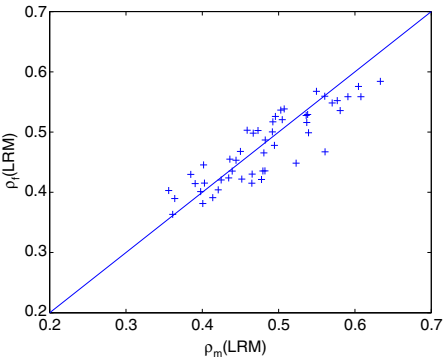
(c) LRM vs SVR-B (Book): $\bar{\rho}_f$(LRM)=0.540, $\bar{\rho}_f$(SVR-B)=0.499, p-value=0.002

the helpfulness correlation between LRM, and SVR-B for HDTV, camera, and book data in few-vote experiments. It is visually apparent that the points in each of these plots primarily scatter above the $y = x$ diagonal line, suggesting that LRM

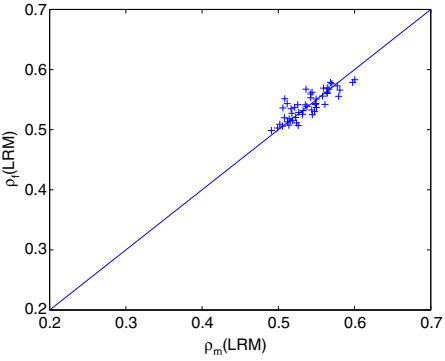
Figure 6 Comparison of the performances of LRM between few-vote data and many-vote data sets



(a) Helpfulness rank correlations of LRM (HDTV): $\bar{\rho}_{\mathrm{m}}$(LRM)=0.619, $\bar{\rho}_{\mathrm{f}}$(LRM)=0.618



(b) Helpfulness rank correlation of LRM (Camera): $\bar{\rho}_{\mathrm{m}}$(LRM)=0.482, $\bar{\rho}_{\mathrm{f}}$(LRM)=0.475



(c) Helpfulness rank correlation of LRM (Book): $\bar{\rho}_{\mathrm{m}}$(LRM)=0.539, $\bar{\rho}_{\mathrm{f}}$(LRM)=0.540

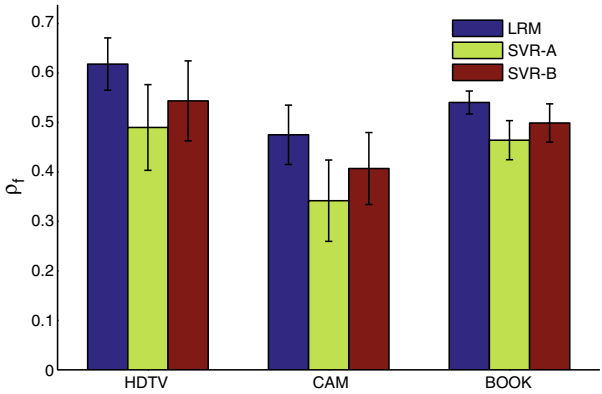


Figure 7 Helpfulness rank correlations of LRM, SVR-A and SVR-B on few-vote data sets. *Color bar*: average of correlation values; *error bar*: standard deviation of correlation values

outperforms SVR-A. This can also be verified by the average of helpfulness rank correlations, $\bar{\rho}$, of the compared algorithms and the p-values of the t-tests (all smaller than 0.005).

Although the proposed LRM algorithm is motivated by the "words of few mouths" phenomena, 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 LRM many-vote experiments for the same set of random partitions and investigate how differently an algorithm performs in few-vote experiments and in many-vote experiments. Figure 6 compares the performances of our algorithm between few-vote and many-vote data sets. It can be seen from (a), (b), and (c) that the scattering of the points in LRM algorithm is tightly around the diagonal line. This indicates that the algorithm is quite robust against "words of few mouths". In particular, the performance of the algorithm under "words of few mouths" and that in absence of "words of few mouths" are quite close, and this similarity in performance is not only in the average sense, but also in the "almost-everywhere" sense.

Figure 7 compiles the performance results of LRM, SVR-A, and SVR-B in few-vote experiments using the HDTV, camera, and book data set. It can be seen from the figure that SVR-B outperforms SVR-A in terms of the mean and variance of 50 experiment results.

Table 1 shows the means and variances of the performances of algorithms. We can observe that the average performance of LRM on few-vote data set is at least 0.04 better than SVR-A and SVR-B.

From these experimental results, we may infer that incorporating probabilistic metrics in the conventional machine-learning based predictors improves the performance of the predictors. Most remarkably, we observe that LRM consistently outperforms SVR, even after SVR is enhanced by the proposed probabilistic helpfulness

Table 1 Mean and stand deviation of the performances of LRM, SVR-A and SVR-B on few-vote data sets

	LRM	SVR-B	SVR-A
HDTV	0.6181 ± 0.0529	0.5436 ± 0.0809	0.4889 ± 0.0866
CAMERA	0.4751 ± 0.0601	0.4069 ± 0.0727	0.3419 ± 0.0822
BOOK	0.5404 ± 0.0232	0.4990 ± 0.0386	0.4642 ± 0.0395

metric, and that LRM suffers little from performance degradation when dealing with "words of few mouths".

Finally, we would like to remark that the proposed LRM algorithm is the most computationally efficient among the three algorithms. Comparing with SVR, LRM runs faster. Also, our LRM algorithm is easy to be extended to an online algorithm.

5 Concluding remarks

Overall, developing a recommender system can often be casted as a machine learning problem [1], and various standard machine-learning toolkits may be applicable for this purpose.

In this paper, we have introduced a widely existing phenomenon, "words of few mouths", in the context of recommender system development based on user opinions. 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 main philosophy advocated in this paper is the use of probabilistic approaches to tackle such challenges, where "words of few mouths" are treated as sparse sampling of some distribution. We proposed a probabilistically formulated metric to improve the performances of the machine-learning based predictors in the presence of "word of few mouths" phenomenon. Furthermore, via developing a logistic regression based learning algorithm for 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".

Although this paper primarily focuses on helpfulness prediction, the general methodology presented in this paper is applicable to the development of the algorithmic engines of other recommender systems from EWOM. In general, probabilistic modeling based inference 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 handling "words of few mouths" in review helpfulness prediction is merely one demonstration of the power of probabilistic algorithms.

This paper uses simple feature models. We note however that probabilistic modeling frameworks in fact allow the use of more sophisticated features. In that case, more complex probabilistic models are required to describe the dependency between different features and the dependency between features of "widgets", users and user opinions. For that purpose, there are several well-known graphical languages, such as Bayesian networks [23] and Markov random fields [19], which one may use to build complex probabilistic models. In addition, there are families of well-established algorithms in these graphical modeling frameworks allowing for a principled approach to developing learning algorithms. (The LRM algorithm developed in this paper is merely one of such examples.)

With this paper, it is our hope that more interest be inspired towards the "words of few mouths" problem and the development of recommender systems can more effectively exploit the information contained in "words of few mouths". (Much of such effectiveness is likely to result from probabilistic models.)

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**, 734–749 (2005)
2. Bertino, E., Ferrari, E., Perego, A.: A general framework for web content filtering. World Wide Web **13**, 215–249 (2009)
3. Bíró, I., Siklósi, D., Szabó, J., Benczúr, A.A.: Linked latent dirichlet allocation in web spam filtering. In: AIRWeb '09: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web, pp. 37–40 (2009)
4. Blei, D.M., Ng, A.Y., Jordan, M.I., Lafferty, J.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 2003 (2003)
5. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery **2**(2), 121–167 (1998)
6. Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines (2001). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
7. Flesca, S., Greco, S., Tagarelli, A., Zumpano, E.: Mining user preferences, page content and usage to personalize website navigation. World Wide Web **8**, 317–345 (2005)
8. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Commun. ACM **35**(12), 61–70 (1992)
9. Han, S.K., Shin, D., Jung, J.Y., Park, J.: Exploring the relationship between keywords and feed elements in blog post search. World Wide Web **12**, 381–398 (2009)
10. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics, pp. 174–181 (1997)
11. Hofmann, T.: Probabilistic latent semantic analysis. In: Proc. of Uncertainty in Artificial Intelligence, UAI' 99, pp. 289–296 (1999)
12. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57 (1999)
13. Hosmer, D.W., Lemeshow, S.: Applied Logistic Regression (Wiley Series in Probability and Statistics), 2nd edn. Wiley-Interscience, New York (2001)
14. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: KDD '04: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177 (2004)
15. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer, New York (2002)
16. Jordan, M.: Why the Logistic Function? A Tutorial Discussion on Probabilities and Neural Networks. Tech. rep., Massachusetts Institute of Technology (1995)
17. Karimzadehgan, M., Zhai, C., Belford, G.: Multi-aspect expertise matching for review assignment. In: CIKM '08: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 1113–1122 (2008)
18. Kim, S.M., Pantel, P., Chklovski, T., Pennacchiotti, M.: Automatically assessing review helpfulness. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 423–430. Association for Computational Linguistics, Sydney (2006)
19. Kindermann, R.: Markov Random Fields and Their Applications (Contemporary Mathematics; vol. 1). American Mathematical Society, Providence
20. Krestel, R., Fankhauser, P., Nejdl, W.: Latent dirichlet allocation for tag recommendation. In: RecSys '09: Proceedings of the Third ACM Conference on Recommender Systems, pp. 61–68 (2009)
21. Liu, Y., Huang, X., An, A., Yu, X.: Modeling and Predicting the Helpfulness of Online Reviews, pp. 443–452 (2008)
22. Lu, Y., Zhai, C., Sundaresan, N.: Rated aspect summarization of short comments. In: WWW '09: Proceedings of the 18th International Conference on World Wide Web, pp. 131–140 (2009)
23. Neapolitan, R.E.: Learning Bayesian Networks. Prentice Hall, Englewood Cliffs (2004)
24. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: EMNLP '02: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, pp. 79–86 (2002)
25. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: CSCW '94: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186 (1994)

26. Schindler, R.M., Bickart, B.: Online Consumer Psychology: Understanding and Influencing Consumer Behavior in the Virtual World. Lawrence Erlbaum, London (2005)
27. Weimer, M., Gurevych, I.: Predicting the perceived quality of web forum posts. In: Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP) (2007)
28. Weimer, M., Gurevych, I., Mühlhäuser, M.: Automatically assessing the post quality in online discussions on software. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 125–128. Association for Computational Linguistics, Prague (2007)
29. Yu, H., Hatzivassiloglou, V.: Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 129–136 (2003)
30. Zhang, Z., Varadarajan, B.: Utility scoring of product reviews. In: CIKM '06: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 51–57 (2006)
31. Zhuang, L., Jing, F., Zhu, X.Y.: Movie review mining and summarization. In: CIKM '06: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 43–50 (2006)