Recommender Systems from "Words of Few Mouths"

Richong Zhang, Thomas Tran, and Yongyi Mao
University of Ottawa
Ottawa, Canada
rzhan025, ttran, yymao@site.uottawa.ca

Abstract

This paper identifies a widely existing phenomenon in web data, which we call the "words of few mouths" phenomenon. This phenomenon, in the context of online reviews, refers to the case that a large fraction of the reviews are each voted only by very few users. We discuss the challenges of "words of few mouths" in the development of recommender systems based on users' opinions and advocate probabilistic methodologies to handle such challenges. We develop a probabilistic model and correspondingly a logistic regression based learning algorithm for review helpfulness prediction. Our experimental results indicate that the proposed model outperforms the current state-of-the-art algorithms not only in the presence of the "words of few mouths" phenomenon, but also in the absence of such phenomena.

1 Introduction

"Electronic word of mouths", or EWOM, on the Internet, may widely refer to information, opinions and user inputs of various kinds, which are provided independently by the Internet users. In general, the problem of developing a recommender system from EWOM may be abstracted in terms of a collection of *"widgets"*, a collection of *users*, and each user's *opinion* 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 opinion of a user on a widget could be in text form (such as a 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 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.

A particular example of a recommender system which we will consider throughout the paper is a "review helpfulness predictor", where each "widget" is a review of a product, and the user opinions are in binary forms, namely, that each user may vote the review `HELPFUL` or `UNHELPFUL`. We will consider the case where there is no information about who votes on which review; 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 based on the existing reviews and the existing votes on those reviews.

It has been recognized that retrieving information from EWOM is often a challenging task. In this paper, we bring to awareness a phenomenon which we call "words of few mouths" (WOFM) which widely exists in EWOM and which amplifies the challenge for developing recommender systems. Specifically the WOFM phenomenon refers to the case where there is a large fraction of "widgets" each only having received opinions from very few users.

The challenges brought by WOFM in the development of recommender system manifest itself as further degraded reliability of user opinions. Using the review helpfulness prediction problem as an example, when each review has been voted by a large number of users, the fraction of positive 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., [Kim *et al.*, 2006; Weimer, 2007; Liu *et al.*, 2008].). However, in the presence of WOFM, the positive vote fraction is a poor indicator of review helpfulness and the performance of the predictors trained this way necessarily degrade.

In general, the negative impact of WOFM 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 WOFM is to remove the "unpopular widgets" (i.e. those receiving few user opinions) 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.

In this paper, we advocate probabilistic approaches to developing recommender system from WOFM, where we use helpfulness prediction as an example. In our approach, instead of considering user opinions on "unpopular" widgets unreliable and throwing them away, we treat user opinions in a probabilistic manner, naturally taking into account the uncertainty arising from WOFM. Specific to the review helpfulness prediction problem, we develop a logistic regression

based algorithm and demonstrate that it significantly outperforms prior arts in this setting. The contributions of this paper are three-fold.

1. We identify and bring to awareness the WOFM phenomenon, a problem widely existing in the development of recommender systems.
2. We propose the use of probabilistic methodologies to tackle such problems.
3. We present an algorithm for a concrete example application and demonstrate its superior performance over existing algorithms. Although our algorithm is based on the logistic regression model for classification, the application we study in this paper does not belong to classification. In particular, in a classification problem, each widget (feature vector) is associated with *one* class label unambiguously, whereas for the helpfulness prediction problem, each widget is associated with a *number* of *possibly contradicting* "class labels" and re-deriving the update equations for the model is necessary. We note that in this paper, we have been brief on many specifics of the review helpfulness prediction problem, for comprehensive details of which, the reader is referred to [Zhang *et al.*, 2011].

We note that the notion of WOFM is closely related to the "Long Tail" phenomenon previously studied in the literature (see, e.g., [Park and Tuzhilin, 2008] and references therein). In the literature, the Long Tail phenomenon refers to the scenarios where a large fraction of sales (or user feedbacks, in the context of this paper) result from the "unpopular" widgets. However, it is worth noting that the notion of "unpopular" in the Long Tail phenomenon may not be consistent with that in WOFM. In particular, the popularity measure of a widget in the Long Tail phenomenon is usually relative, namely, via comparing with other widgets; an unpopular widget in that setting may still be associated with a significant amount of user feedback and presents little difficulty for developing predictors, whereby disqualifying themselves as unpopular widgets in the context of WOFM. The literature of recommender systems dealing with Long Tails are primarily concerned with developing techniques to handle the non-uniformity of feedback data (e.g., [Park and Tuzhilin, 2008] separates "Tail widgets" and "Head widgets" and treats them differently).

We also like to stress that although the presented logistic regression algorithm in this paper demonstrates significant advantages over other algorithms, we have no intention to mean that this is the only algorithm fitting the probabilistic methodology we advocate. Indeed, in [Zhang and Tran, 2011], we have presented EM-based probabilistic algorithm for review helpfulness prediction. However the performance of the EM-based algorithm is in fact inferior to the algorithm in this paper when applied to this setting (data not shown).

2 Probabilistic Approach to Developing Recommender Systems

Overall, developing a recommender system can often be casted as a machine learning problem [Adomavicius and Tuzhilin, 2005], and various standard machine-learning toolkits may be applicable for this purpose. Here we advocate a probabilistic modeling approach, particularly in the case of "words of few mouths". Methodologically, we first create a probabilistic model, or a family of hypotheses, to characterize the statistical dependency between the "widgets", 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 data set in some well-principled sense of optimality. To be more concrete, the remainder of this paper focuses on developing algorithmic engines for review helpfulness predictor.

2.1 Probabilistic Formulation of Helpfulness Prediction 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 votes on review i and the negative 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 $\{J(i) : i \in I\}$.

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 follows.

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

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 attempts 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 not suitable for the case of WOFM since the positive vote fraction, as an indicator of helpfulness, suffers greatly from statistical irregularity.

We now present a probabilistic approach to the helpfulness prediction problem. 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 $\mathcal{D} \times \mathcal{R}$ according to the following procedure:

1. There is an unknown distribution $p_{\mathbf{D}}$ on $\mathcal{D}$; applying i.i.d. sampling of $\mathcal{D}$ under $p_{\mathbf{D}}$ results in d_I.
2. For each $d \in \mathcal{D}$, there is an unknown conditional distribution $p_{\mathbf{R}|d}$ on $\mathcal{R}$. For each $d_i, i \in I$, applying i.i.d. sampling of $\mathcal{R}$ under $p_{\mathbf{R}|d_i}$ and let the drawn rating functions act on d_i result in the set of votes $v_{J(i)}$ on review 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.

Under the above generative interpretation of data $(d_I, v_J, \{J(i) : i \in I\})$, we characterize the helpfulness 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. Noting 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.

This allows a probabilistic formulation of helpfulness prediction problem: Given data $(d_I, v_J, \{J(i) : i \in I\})$ generated from the above procedure, determine the distribution $p_{\mathbf{V}|\mathbf{D}}$.

Although one may consider various options to adapt a classification methodology to solving 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

$$p^*_{\mathbf{V}|\mathbf{D}} := \operatorname*{argmax}_{p_{\mathbf{V}|\mathbf{D}} \in \Theta_{\mathbf{V}|\mathbf{D}}} \log p_{V_J|D_I}(v_J|d_I) \tag{2}$$

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

$$p^*_{\mathbf{V}|\mathbf{D}} = \operatorname*{argmax}_{p_{\mathbf{V}|\mathbf{D}} \in \Theta_{\mathbf{V}|\mathbf{D}}} \sum_{i \in I} \sum_{j \in J(i)} \log p_{\mathbf{V}|\mathbf{D}}(v_j|d_i) \tag{3}$$

As is common in many machine-learning problems, the huge dimensionality of space $\mathcal{D}$ makes solving problem (3) infeasible. A wide-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 (3) is then modified to finding

$$p^*_{\mathbf{V}|\mathbf{F}} = \operatorname*{argmax}_{p_{\mathbf{V}|\mathbf{F}} \in \Theta_{\mathbf{V}|\mathbf{F}}} \sum_{i \in I} \sum_{j \in J(i)} \log p_{\mathbf{V}|\mathbf{F}}(v_j|f_i), \tag{4}$$

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 [Hosmer and Lemeshow, 2000] developed in classification literature.

2.2 Logistic Regression for Helpfulness Prediction

Central to solving the optimization problem specified in (4) 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*. More precisely, we define

$$p_{\mathbf{V}|\mathbf{F}}(1|f) = \mu(\lambda), \tag{5}$$

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

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

and $\lambda := \theta^T f$ for some vector θ having the same dimension as feature vector f.We note that since $p_{\mathbf{V}|\mathbf{F}}(1|f) + p_{\mathbf{V}|\mathbf{F}}(0|f) = 1$, Equation (5) completely defines model $\Theta_{\mathbf{V}|\mathbf{F}}$, namely,

$$\Theta_{\mathbf{V}|\mathbf{F}} := \{p_{\mathbf{V}|\mathbf{F}} \text{ satisfying } p_{\mathbf{V}|\mathbf{F}}(1|f) = \frac{1}{1 + e^{-\theta^T f}} : \theta \in \mathbb{R}^m\}, \tag{6}$$

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 conditional 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.

Now using model $\Theta_{\mathbf{V}|\mathbf{F}}$ defined in Equation (6), the optimization problem of Equation (4) reduces to solving

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^m}{\operatorname{argmax}} \sum_{i \in I} \sum_{j \in J(i)} \left[v_j \log \mu(f_i) + (1 - v_j) \log \left(1 - \mu(f_i)\right) \right], \tag{7}$$

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

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

$$\frac{dl}{d\theta} = \sum_{i \in I} \sum_{j \in J(i)} f_i (v_j - \mu(f_i)) \tag{8}$$

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

$$\theta^{t+1} := \theta^t + \rho \sum_{i \in I} \sum_{j \in J(i)} f_i (v_j - \mu(f_i)). \tag{9}$$

where ρ is a choice of step size.

3 Experimental Evaluation

To demonstrate the effectiveness of the proposed approach, we experimentally evaluate our logistic regression model (LRM) and compare it with two most well-known machine learning methods, Support Vector Regression (SVR) [Burges, 1998], and Artificial Neural Network (ANN) [Bishop, 1996], in review helpfulness prediction. This section presents our method of evaluation, experimental setups and results of comparison.

3.1 Method of Evaluation

A difficulty associated with "words of few mouths" in evaluating the performances of algorithms is the lack of benchmarks for "unpopular widgets". In the context of helpfulness prediction, this difficulty translates to the question what to use as the 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 $M = 10$.

We construct a "few-vote" data set from the many-vote data set by randomly selecting k user's votes for each review and removing all other votes. Given the value k, a few-vote data set may also be referred to as a k-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 exactly k users. In our study, we focus on the case of $k = 5$.

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 and total of 50 random partitions $(\mathcal{N}, \mathcal{T})$'s are generated.

In this setting, two types are experiments are 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.

Helpfulness rank correlation is used as the metric in our study to evaluate the performance of compared algorithms. It is essentially the Spearman's rank correlation coefficient η between the helpfulness ranks of the testing reviews predicted by an algorithm and that according to the corresponding positive vote fractions.

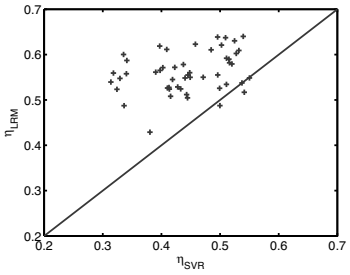
$$\eta = 1 - \frac{6 \sum_{j \in \mathcal{T}} (x_j - y_j)^2}{|\mathcal{T}|(|\mathcal{T}|^2 - 1)}, \tag{10}$$

where x_j is the rank of review j according to helpfulness predicted by an algorithm and y_i is the rank of review j according to the positive vote fraction of review j obtained from the many-vote data set. The average $\bar{\eta}$ of helpfulness rank correlations may be computed across all random partitions to obtain the overall performance of an algorithm. In addition, the correlation values (η's) can be used in a t-test to determine whether an algorithm performs significantly differently from another algorithm.

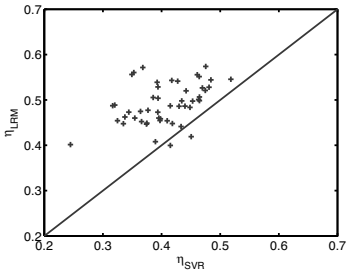
3.2 Experimental Setup

As there are no standard customer review corpus available, we utilize the web services provided by Amazon.com to crawl the web site and obtain two data sets of review documents and vote information: HDTV data set and digital camera data set. The HDTV many-vote data set contains 14,397 votes and 583 reviews and the camera many-vote data set contains 13,826 votes and 906 reviews.

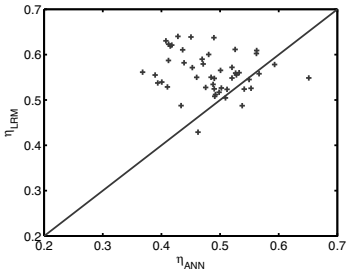
Since the objective of this work is not to develop a sophisticated language model but rather to study the WOFM problem, we use the "bag of words" language model to represent each review document.For each partition $(\mathcal{N}, \mathcal{T})$, prior to the training of the three algorithms, dimensionality reduction is performed using the principal component analysis (PCA). We select the top 200 principal components in PCA, which accounts for 70% of the total variance. We implement a three-layer back-propagation (BP) ANN. The number of neurons in the hidden layer is chosen to be 10. Each node utilizes



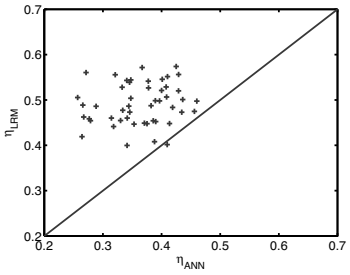
(a) LRM vs SVR (HDTV): $\bar{\eta}_{\mathrm{LRM}} = 0.561$, $\bar{\eta}_{\mathrm{SVR}} = 0.444$, p-value=5.66e-16



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



(c) LRM vs ANN (HDTV): $\bar{\eta}_{\mathrm{LRM}} = 0.561$, $\bar{\eta}_{\mathrm{ANN}} = 0.472$, p-value=3.95e-5



(d) LRM vs ANN (Camera): $\bar{\eta}_{\mathrm{LRM}} = 0.491$, $\bar{\eta}_{\mathrm{ANN}} = 0.35$, p-value=1.25e-13

Figure 1: Comparison of helpfulness rank correlation using 5-vote data sets.

sigmoid transfer function. The training of the ANN is terminated after 1000 training iterations or when the error term is less than 0.001. We also implement a SVR algorithm using the LibSVM [Chang and Lin, 2001] 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. The learning targets for both ANN and SVR are chosen to be the positive vote fractions of the training reviews.

3.3 Experimental Results

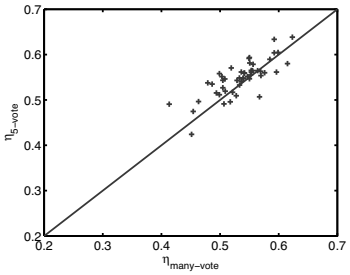
Figure 1 shows a set of scatter plots that compare the helpfulness rank correlation between LRM, SVR, and ANN for HDTV and camera data in few-vote experiments. Each point in any plot corresponds to one partition $(\mathcal{N}, \mathcal{T})$. It is visually apparent that the points in each of these plots primarily scatter above the $y = x$ diagonal line, 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 correlations, $\bar{\eta}$, 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 WOFM, 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 similar performance advantage of LRM as those shown in Figures 1 are obtained (data not shown). It is of interest to compile the results obtained in the two set of experiments and investigate how differently an algorithm performs in few-vote experiments and in many-vote experiments. Figure 2 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 LRM algorithm is tightly around the diagonal line. This indicates that the algorithm is quite robust against WOFM. In particular, the performance of the algorithm under WOFM and that in absence of WOFM are 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 WOFM. Under WOFM scenarios, not only the average performance degrades, the performances of SVR and ANN also severely suffer from large stochastic variations. This is expected, since using positive vote fractions as the training target necessarily suffer from significant statistical irregularity induced by WOFM. Finally, we would like to remark that the proposed LRM algorithm is the most computationally efficient among the three algorithms.

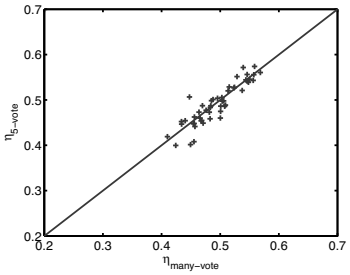
4 Concluding Remarks

In this paper, we have introduced a widely existing phenomenon, "words of few mouths", in the context of recommender system 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 unutilized, leading to lack of resources for learning, or becoming an additional source of "noise" in the training of algorithms.

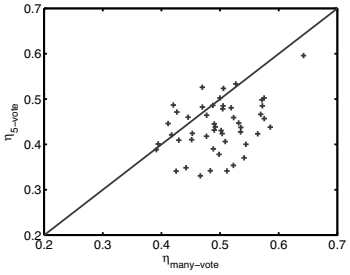
The main philosophy advocated in this paper is the use



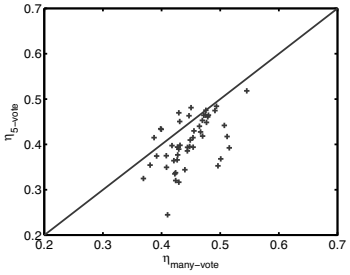
(a) Rank correlations of LRM (HDTV): $\bar{\eta}_{\mathrm{many-vote}} = 0.535$, $\bar{\eta}_{5-\mathrm{vote}} = 0.548$



(b) Rank correlation of LRM (Camera): $\bar{\eta}_{\mathrm{many-vote}} = 0.494$, $\bar{\eta}_{5-\mathrm{vote}} = 0.491$



(c) Rank correlation of ANN (HDTV): $\bar{\eta}_{\mathrm{many-vote}} = 0.497$, $\bar{\eta}_{5-\mathrm{vote}} = 0.428$



(d) Rank correlation of SVR (Camera): $\bar{\eta}_{\mathrm{many-vote}} = 0.447$, $\bar{\eta}_{5-\mathrm{vote}} = 0.406$

Figure 2: Comparison of the performances of each algorithm between 5-vote data and many-vote data.

of probabilistic approaches to tackle such challenges, where WOFM is treated as sparse sampling of some distribution. 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 WOFM.

Although this paper primarily focuses on helpfulness prediction, the general methodology presented is applicable to the algorithmic engines of other recommender systems from EWOM. Our results suggest that probabilistic modeling based inference and learning algorithms are particularly suitable for handling uncertainty, errors and missing information in the data set.

References

[Adomavicius and Tuzhilin, 2005] Gediminas Adomavicius and Alexander 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.

[Bishop, 1996] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1 edition, January 1996.

[Burges, 1998] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovory*, 2(2):121–167, 1998.

[Chang and Lin, 2001] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*.

[Hosmer and Lemeshow, 2000] David W. Hosmer and Stanley Lemeshow. *Applied logistic regression*. Wiley-Interscience Publication, 2 edition, 2000.

[Kim *et al.*, 2006] Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*,2006.

[Liu *et al.*, 2008] Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. Modeling and predicting the helpfulness of online reviews. In *Proceedings of the 18th IEEE International Conference on Data Mining (ICDM)*,2008.

[Park and Tuzhilin, 2008] Yoon-Joo Park and Alexander Tuzhilin. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM Conference on Recommender Systems, (RecSys)*, 2008.

[Weimer, 2007] Iryna Weimer, Markus; Gurevych. Predicting the perceived quality of web forum posts. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP)*,2007.

[Zhang and Tran, 2011] Richong Zhang and Thomas T. Tran. A Helpfulness Modeling Framework for Electronic Word-of-Mouth on Consumer-Opinion Platforms. *ACM Transaction on Intelligent Systems and Technology*, 2(3), 2011.

[Zhang *et al.*, 2011] Richong Zhang, Thomas T. Tran and Yongyi Mao. Opinion Helpfulness Prediction in the Presence of 'Words of Few Mouths'. Submitted to *World Wide Web*.