Generating Virtual Ratings from Chinese Reviews to Augment Online Recommendations

WEISHI ZHANG and GUIGUANG DING, Tsinghua University
LI CHEN, Hong Kong Baptist University
CHUNPING LI and CHENGBO ZHANG, Tsinghua University

Collaborative filtering (CF) recommenders based on User-Item rating matrix as explicitly obtained from end users have recently appeared promising in recommender systems. However, User-Item rating matrix is not always available or very sparse in some web applications, which has critical impact to the application of CF recommenders. In this article we aim to enhance the online recommender system by fusing virtual ratings as derived from user reviews. Specifically, taking into account of Chinese reviews' characteristics, we propose to fuse the self-supervised emotion-integrated sentiment classification results into CF recommenders, by which the User-Item Rating Matrix can be inferred by decomposing item reviews that users gave to the items. The main advantage of this approach is that it can extend CF recommenders to some web applications without user rating information. In the experiments, we have first identified the self-supervised sentiment classification's higher precision and recall by comparing it with traditional classification methods .Furthermore, the classification results, as behaving as virtual ratings, were incorporated into both user-based and item-based CF algorithms. We have also conducted an experiment to evaluate the proximity between the virtual and real ratings and clarified the effectiveness of the virtual ratings. The experimental results demonstrated the significant impact of virtual ratings on increasing system's recommendation accuracy in different data conditions (i.e., conditions with real ratings and without).

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

General Terms: Algorithms, Performance, Experimentation

Additional Key Words and Phrases: Information retrieval, online recommendation, sentiment analysis

ACM Reference Format:
Zhang, W., Ding, G., Chen, L., Li, C., and Zhang, C. 2013. Generating virtual ratings from Chinese reviews to augment online recommendations. ACM Trans. Intell. Syst. Technol. 4, 1, Article 9 (January 2013), 17 pages.
DOI = 10.1145/2414425.2414434 http://doi.acm.org/10.1145/2414425.2414434

1. INTRODUCTION

Recommender systems that suggest unknown interesting items to users have been developed rapidly in recent years, among which collaborative filtering (CF) recommenders

This research was supported by the National Basic Research Project of China (Grant No. 2011CB707000), the National Natural Science Foundation of China (Grant No. 60972096, 90924003), and HKBU FRG2/10-11/041.
Authors' addresses: W. Zhang, G. Ding, C. Li, and C. Zhang, School of software, Tsinghua University, Beijing 100084, China; email: zhang-ws08@mails.tsinghua.edu.cn, {dinggg,cli}@tsinghua.edu.cn, zhangchengbodragon@163.com; L. Chen, Department of Computer Science, Hong Kong Baptist University, Hong Kong, email: lichen@comp.hkbu.edu.hk.

DOI 10.1145/2414425.2414434 http://doi.acm.org/10.1145/2414425.2414434

are those of the broadly applied approaches. The CF approaches principally derive recommendations for a user based on the preferences of other users who were discovered with similar tastes [Breese et al. 1998]. Indeed, most CF-based systems reply on item ratings as explicitly obtained from end users for the calculation of user-user or item-item similarity. However, few of them investigated the potential effect of user reviews (or called textual comments to items), as another type of valuable user-generated sources, on boosting the recommendation accuracy and addressing rating sparsity limitations [Papagelis et al. 2005]. In our work, after surveying existing popular Chinese media-sharing Web sites, such as Youku,[1] Ku6[2] and Tudou,[3] we found that they all possess large amounts of review data that visitors have written for expressing their opinions. The question is then whether/how we could incorporate user reviews into CF-based systems, so as to significantly augment recommendations, especially in the condition that users' real ratings are not available. In fact, though above mentioned sites enable users to thumb up/down for an item (i.e., providing binary rating to an item), they did not record who did this action. So it is hard to rely on this kind of rating info to obtain users' preferences, while reviews could be potentially more helpful as they were all attached with user IDs.

To achieve the goal of incorporating user reviews, we have particularly attempted to derive "virtual ratings" from user reviews through the method of sentiment classification, so that given a piece of text, its latent opinion (represented by sentiment polarity such as positive, neutral, or negative), can be discovered for reflecting the user's preference on the corresponding item. Hereafter, we call such ratings that are derived from user reviews as "virtual ratings," to be conceptually different from the user inputted ratings.

More specifically, our system is targeted to provide review-based recommendations for Chinese sites. Thus, we have first investigated Chinese reviews' characteristics, which include: (1) each user review is usually short and noisy (including advertisements, hyperlink text etc.); (2) many reviews contain emoticons such as smiley faces (in our experimental data, 41% reviews have this property); (3) the ratio between positive and negative reviews is not 1:1., though most of related sentiment classification methods are under this 1:1 assumption [Blitzer et al. 2005; Turney 2002; Zagibalov and Carroll 2008a].

Taking into account these characteristics, we have been aiming at developing the SElf-Supervised, Lexicon-based and Corpus-based (SELC) model, which is a self-supervised sentiment classification approach to determining the overall sentiment polarity of a review document that contains both textual words and emoticons. As a result from the model, we use the sentimental polarities of reviews as one kind of resources for recommendation. In the experiments, we have first identified our sentiment classification method's higher precision and recall by comparing it with other typical classification methods. Second, we tested the fusion effect of virtual ratings in two datasets: one is without users' real ratings, and the virtual ratings were incorporated into user-based and item-based CF algorithms respectively to evaluate which fusion mechanism can better exploit the virtual ratings' merit; another is with users' real ratings, and the virtual ratings are evaluated by a matching experiment, then both of the virtual ratings and real ratings are used to predict recommendations to get a comparable results over the user-based and item-based approaches.

The rest of this article is organized as follows. We first introduce related work in Section 2, and then given the overview of our approach that is divided into two phases

[1]www.youku.com.
[2]www.ku6.com.
[3]www.tudou.com.

(Section 3). Sections 4 and 5 give the detailed algorithm for each phase, followed by Section 6 with experimental procedures and results analysis. Finally, we conclude this article and indicate its future directions.

2. RELATED WORK

2.1. Sentiment Classification

In supervised sentiment classification methods, standard machine learning techniques such as Support Vector Machine (SVM) and Naive Bayes have been usually used [Pang et al 2002; Alpaydin 2004]. Different factors affecting the machine learning process are investigated. For example, linguistic, statistical, and n-gram features were researched in Dave et al. [2003]. Selected words and negation phrases were investigated in Na et al. [2004]. However, the performance of supervised approaches normally decreases when training data is insufficient [Aue and Gamon 2005; Read 2005].

On the contrary, unsupervised approaches make the assumption that there are certain words people tend to use to express strong sentiment, so that they might suffice to classify the documents. In Turney [2002], an unsupervised sentiment classification approach was proposed by calculating the mutual information between each phrase in a document and the selected two seed words: excellent and poor. Fewer seed words imply less domain-dependency. Zagibalov and Carroll [2008a] only assign one word *good* as a seed positive word, and use negation words such as "not" to find initial negative expressions. In Zagibalov and Carroll [2008b], even the one word "good" is ignored, and seed words are automatically generated based on a linguistic pattern (called "negated adverbial construction") like "not very good". Experimental results show that this method achieves similar performance to supervised methods on Chinese product reviews.

SELC Model (SElf-Supervised, Lexicon-based and Corpus-based Model) [Qiu et al. 2009] is proposed for self-supervised sentiment classification on Chinese IT product reviews. The model includes two submodels. In the first phase, some reviews are initially classified based on a sentiment dictionary. Then more reviews are classified through an iterative process with a negative/positive ratio control. In the second phase, a supervised classifier is learned by taking some reviews classified in the first phase as training data. Then the supervised classifier applies on other unclassified reviews to revise the results produced in the first phase. In this article, we improve this work by considering the special features of real Chinese online reviews and use the sentimental polarities of reviews as resources for recommendations.

2.2. Recommender Systems

Since 1990s, recommender systems have been explored in many product domains, that is, movies [Christakou and Stafylopatis 2005], TVs [Setten and Veenstra 2003], Web pages [Balabanovic 1998] with the objective of recommending items matched to users' profiles [Yang et al. 2007]. In recent years, much more techniques have been developed in recommender systems in order to derive better performance [Gunawardana and Meek 2009; de Gemmis et al. 2008; TsoSutter et al. 2008]. However, most of works are limited when user preference data (i.e., ratings) are hardly obtainable from real sites (e.g., the video-sharing sites). To address this limitation, tags (in form of user-defined keywords) have been utilized as supplementary source to predict user interests [Tso-Sutter et al. 2008]. Tso-Sutter et al. [2008] proposed a generic method that allows tags to be incorporated into standard CF algorithms, by reducing the three-dimensional correlation to three two-dimensional correlations and then applying a fusion method to reassociate these correlations. de Gemmis et al. [2008] have developed a strategy to infer user interests by applying machine learning techniques to learn from both

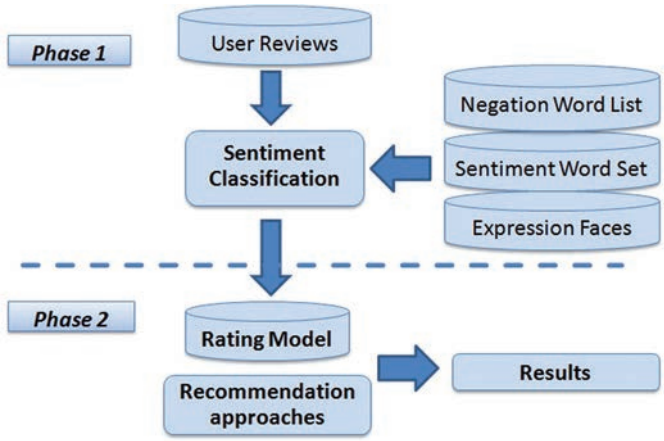

Fig. 1. Proposed online recommender algorithms for Chinese sites.

the "official" item descriptions provided by a publisher, and tags that users used to annotate relevant items.

Matrix factorization based techniques have proven to be efficient in recommender systems when predicting user preferences from known user-item ratings. Paterek applied successfully various matrix factorization techniques [Paterek 2007] by adding biases to the regularized MF, post processing the residual of MF with kernel ridge regression, using a separate linear model for each movie, and by decreasing the parameters in regularized MFs. Kurucz et al. [2007] showed the application of expectation maximization based MF methods for Netflix prize. Recently, several matrix factorization methods [Salakhutdinov and Mnih 2008a, 2008b] have been proposed for collaborative filtering. These methods all focus on fitting the user-item rating matrix using low-rank approximations, and use it to make further predictions.

However, to the best of our knowledge, only a few papers have considered user reviews and integrated their sentiment analysis results into the generation of recommendations. Leung et al. [2006] have attempted to identify features from reviews to infer ratings, but after no detailed description of how the method was implemented. The sentiment analysis approaches in Ganu et al. [2009] are supervised and hence need manually annotated training data. Jakob et al. [2009] proposed three approaches to extract movie aspects as opinion targets and use them as features for the collaborative filtering on IMDB dataset. Each of these approaches requires different amounts of manual interaction. However, none of the prior papers has explored the combination of virtual ratings and review sentiment analysis on a Chinese dataset. Our work exerts to address this limitation by proposing a self-supervised sentiment classification approach and applying the results to predict the virtual ratings on items, so as to be effectively fused into standard CF algorithm in a Chinese dataset.

3. OVERVIEW OF OUR APPROACH

Our recommender algorithm is proposed to study the roles of online reviews in augmenting recommenders in current Chinese media-sharing sites. The algorithm concretely consists of two phases: (1). SElf-Supervised, Lexicon-based and Corpus-based Model (SELC) for Review Sentiment Classification and (2). Item Recommendation. Figure 1 shows the algorithm flow. Phase 1 and Phase 2 are separated by a dash line.

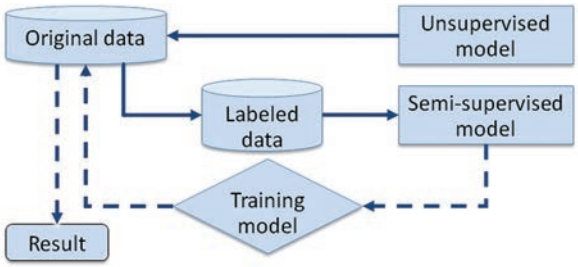

Fig. 2. Flow chart of the self-supervised sentiment classification process.

4. PHASE 1: SELF-SUPERVISED REVIEW SENTIMENT CLASSIFICATION

Based on a *sentiment word set*, a *negation word list*, and an *emoticon set*, Phase 1 uses a self-supervised approach (SELC) to identify the sentiment polarity of each review. Figure 2 shows the flow chart of the whole self-supervised sentiment classification process.

It concretely consists of two models, that is, unsupervised model and semi-supervised model. In the unsupervised model, an unsupervised approach is applied on the original data to automatically label some data. In the semi-supervised model, a semi-supervised approach is applied on the labeled data to acquire a training model. Finally, the model is applied on the original data to do the sentiment classification. In Figure 2 the solid lines refer to the unsupervised model while dash ones refer to the supervised model.

4.1. Unsupervised Model

In the unsupervised model of our self-supervised, lexicon-based, and corpus-based (SELC) modeling, a sentiment vocabulary is initialized by a general sentiment dictionary. The vocabulary is used to label reviews. Then more sentiment words are found from the labeled reviews for updating the vocabulary. The new vocabulary then helps classify more reviews. By this iterative process, the vocabulary and labeled reviews are updated and enlarged step by step. In the iterative process, the positive/negative ratio is controlled. The algorithm ranks the reviews during each iteration and keeps the same number of top-ranked positive and negative reviews. Additionally, the emoticon scoring analysis is integrated into the iterative process of the unsupervised model to get more accurate results. Specifically, the unsupervised model consists of following steps:

4.1.1. Step 1: Initializing Sentiment Element Sets. The *sentiment element sets* consist of two sets, that is, *sentiment word set* and *emoticon set*. The sentiment word set, denoted by W_{sen}, includes a list of word items, each of which is assigned with a sentiment score. W_{sen} is initialized by a general sentiment dictionary, which usually includes a lot of positive and negative words. A positive word is initially assigned with score $+1.0$, while a negative word is assigned with score -1.0. Monosyllabic words are filtered from W_{sen}, because most of them are too ambiguous to provide reliable sentiment. In addition, since the general sentiment dictionary is applicable to many domains, this method has the potential to be domain independent.

The emoticon set, denoted by E, is the set of the emoticons (e.g., smiley or sad faces) used by the users to express their preferences. Because the emoticons are widely used by the users in many resource-sharing Web sites to express their opinions, they play an important role in the task of our item review sentiment classification. First, we manually remove all of the nonesentiment-bearing emoticons, for instance, [Oh . . .] and [Well . . .], from the whole set of crawled expression emoticons. Then we add the remaining part into E and according to the sentiment they express, the emoticons

are divided into two kinds: positive and negative. Each positive face in E is initially assigned with score $+1.0$, and a negative emoticon is assigned with score -1.0. We selected 10 positive emoticons and 5 negative emoticons as the initial emoticon set (see details in the experiment Section 6.2.1).

For the generation of the negation word list, we manually selected ten most frequently used negation words, such as "不" ('not'), "不会" ('would not'), "没有" ('don't have'), "没" ('don't have'), etc. (see the dataset used in the experiment Section 6.2.1).

4.1.2. Step 2: Identifying Review Sentiment Scores. Through analyzing online reviews, two kinds of reviews have been found in most of view-sharing Web sites: users expressed their opinions on the items, or users expressed their opinions on other users' reviews. We call the first kind as *item-oriented reviews* and the second kind as *user-oriented reviews*. It's easy to differentiate between these two kinds of reviews, since the user-oriented reviews always start with a "[reply to] + [other user]" writing styles. Since the sentiment of user-oriented reviews is usually not directly related to the items, we only focus on item-oriented reviews. We also removed some noise data including advertisements and hyperlink text.

Therefore, at first, a preprocessing was conducted to filter out all the user-oriented reviews and noise data according to their writing styles. Given an item i, all of its related reviews are denoted by $Rev(i)$. Each review r ($r \in Rev(i)$) is then divided into clauses by punctuation marks.

Secondly, for each clause, if it contains sentiment word items as appearing in W_{sen} (the sentiment word set), each sentiment word item w of the clause is scored by Equation (1), where L_w is the length of the word item, L_{clause} is the length of the clause, S_w^W is the word item's current sentiment score in W_{sen}, and N_w is a negation check coefficient that has a default value of 1.0. If the word item is preceded by a negation within the specified zone, N_w is set to -1.0. We have two assumptions to design Equation (1): (1) at the most time the longer the length of a Chinese word is, the clearer its sentiment polarity is (the square of L_w is to enlarge the assumption); (2) the same word in a shorter clause usually expresses stronger sentiment than that in a longer clause.

$$S_w = \frac{L_w^2}{L_{clause}} S_w^W N_w. \tag{1}$$

Then the sentiment score of a clause c, denoted by $CS(c)$, is calculated by $CS(c) = \sum S_w$ for all $w \in c$. For each review r, the *ReviewWordScore* (the sentiment score of a review taking into account of its contained sentiment words), denoted by $RS^W(r)$, is subsequently calculated according to Equation (2).

$$RS^W(r) = \sum_{c \in r} CS(c). \tag{2}$$

For review r, the *ReviewEmoticonScore*, denoted by $RS^E(r)$, is also calculated according to Equation (3), where S_e^E is the current sentiment score of emoticon item e as appearing in E.

$$RS^E(r) = \sum_{e \in F_{sen} \cap e \in r} S_e^E. \tag{3}$$

Finally, the sentiment score of the review r, denoted by $RS(r)$ can be computed using:

$$RS(r) = \alpha RS^W(r) + (1 - \alpha) RS^E(r), \tag{4}$$

where parameter $\alpha \in [0,1]$ determines the weight put on each factor, that is, the balance between the review's word sentiment score $RS^W(r)$ and its emoticon sentiment score $RS^E(r)$.

4.1.3. The Bias Caused by the Missing of Ratio Control. Basically, after this step, a review r will be classified as positive (if $RS(r) > 0$) or negative (if $RS(r) < 0$). This policy looks good but would cause sentiment bias for the following step of updating the sentiment sets (Step 4). In order to explain the bias in a better way, we skip Step 3 and introduce Step 4 first.

4.1.4. Step 4: Updating the Sentiment Element Sets. In Step 4, the sentiment word set W_{sen} and *emoticon set* E are to be updated (and usually enlarged).

For sentiment word set W_{sen}, each lexical item[4] that occurs at least twice in those classified reviews is taken as a candidate word item. For an candidate word item w, denote the number of positive reviews containing w as N_w^p, and the number of negative reviews containing w as N_w^n (preceded by a negation will make the account reduce by one, and N can be negative). The idea of updating sentiment word set W_{sen} is then: if N_w^p is much bigger than N_w^n, w is very likely to be a positive word item, and vice versa. The following formula is used to be the measure.

$$difference(w) = \frac{\left|N_w^p - N_w^n\right|}{\left(N_w^p + N_w^n\right)}. \tag{5}$$

If *difference(w)*$\geq$1, w is included in W_{sen} (current items in W_{sen} will be removed if they no longer satisfy this condition). The sentiment score of w in W_{sen} is updated as

$$S_w^W = N_w^p - N_w^n. \tag{6}$$

For updating the emoticon set, for an emoticon item e, denote the number of positive reviews containing e as N_e^p, and the number of negative reviews containing e as N_e^n. The idea of updating emoticon set E is similar to the sentiment word set updating: if N_e^p is much bigger than N_e^n, then e is very likely to be a positive emoticon item, and vice versa. The following formula is the measure.

$$difference(e) = \frac{\left|N_e^p - N_e^n\right|}{\left(N_e^p + N_e^n\right)}. \tag{7}$$

If *difference(e)* $\geq$ 1, w is included in E(current items in E will be removed if they no longer satisfy this condition). The sentiment score of w in E is updated as

$$S_e^E = N_e^p - N_e^n. \tag{8}$$

4.1.5. Step 3: Classifying Reviews based on Ratio Control. After the introduction of Step 4, we can give an example to explain the bias in Step 4 caused by the missing of ratio control. If there are 20 reviews classified as positive and 10 reviews as negative, then the number of words only occurring in the positive reviews is more likely to be bigger than the number of words only occurring in negative ones. If the word "screen" only occurs in one of the positive reviews, then "screen" will be assigned with a sentiment score of 1.0 (Step 4 will explain how the score 1.0 is obtained using Equation (6)), and therefore be judged as a positive word item. But in fact, such a word may not have any sentiment polarity. Such bias is caused by unequal number of positive and negative documents. To overcome the bias, a ratio control is designed, which requires the number of positive and negative reviews in the classified sentiment review list to be the same.

Denote the number of positive and negative reviews in one round of iteration as $RN_{positive}$ and $RN_{negative}$ respectively. To realize the ratio control, first, rank all reviews

[4]Let N be the length of a zone, a lexical item is a sequence of Chinese characters excluding punctuation marks, from unigram to N-gram, in an enclosing zone.

1. Let RN_{min}=$Min(RN_{positive}, RN_{negative})$.
2. Rank all reviews in descending order by their *RS*.
3. Document labeling:
   - 3.1 Label the top RN_{min} reviews in the ranking list as positive.
   - 3.2 Label the tail RN_{min} reviews in the ranking list as negative.
   - 3.3 Others are left unlabeled.

Fig. 3. Review sentiment classification with ratio control.

according to their sentiment scores $RS(r)$. Second, take the smaller value between $RN_{positive}$ and $RN_{negative}$, that is, $Min(RN_{positive}, RN_{negative})$, as a threshold, and remain the positive and negative documents above the threshold in the sentiment review list, and remove others.

Figure 3 shows the process of classifying reviews with ratio control. Those reviews form the sentiment review list.

4.1.6. Step 5: Iteration Control. The unsupervised approach iterates from Step 1 to Step 4. In the SELC model, the iteration completes when β % of documents have been labeled. Through the empirical test in one of our prior works [Zhang et al. 2009], we found that the optimal value for β is 0.618 for well balancing both accuracy and time efficiency. That is, when 61.8% of documents have been labeled, the iteration procedure can be completed and the labeled documents can be used as the training data for the semi-supervised model.

4.2. Semi-Supervised Model

In semi-supervised model, the Support Vector Machine (SVM) classifier with a linear kernel is selected. Specifically, in this model, the items of sentiment element sets as retrieved from the last iteration are used as the feature set. TFIDF measure (see Equation (9)) is used to compute weights for the items in both sentiment word set and emoticon set.

$$w_i = tf_i \times \log \frac{N_i}{df_i}. \tag{9}$$

Then the SVM classifier applies the data to do the classification and get the final review sentiment classification results.

5. PHASE 2: ITEM RECOMMENDATION

When Web sites do not support users to give ranking scores for items, we cannot get the real ratings from users. In this condition, virtual ratings as derived from the reviews will be fully incorporated and behave as primary resource for producing recommendations.

Because after the process of Phase 1, each review r will be classified as *positive, negative* or *neutral*, at this step, we use the review sentiment classification results to predict the virtual rating matrix, which can be then taken as input to the standard collaborative filtering recommender algorithms (user-based and item-based).

In R_{UI}, each user has a virtual *User-Item Vector*, that is, V_{UI} *(u)*. Each V_{UI} consists of three parts that is, *Like*+, *Dislike*− and *Unknown*. The *Like*+ part of the V_{UI} consists of the items liked by the user u (positive and neutral ones[5]), while the *Dislike*− and

[5]According to the habits of the majority online users, if they are interested in the item, they are likely to give reviews for it even if the polarities of reviews are not clear sometimes (neutral).

Unknown parts consist of the items disliked or unknown to user u (negative and unknown ones) respectively.

Firstly, given an item i and a user u, the set of all the reviews that user u puts on item i is denoted as *Rev(u,i)*. The set of all the positive reviews in *Rev(u,i)* is denoted as $Rev(u,i)^{pos}$, while the set of all the negative reviews in *Rev(u,i)* is denoted as $Rev(u,i)^{neg}$.

Then, for a user u, we calculate the sets of $Rev(u,i)^{pos}$ and $Rev(u,i)^{neg}$ for all the items. and build the *User-Item Vector* (V_{UI} *(u)*) of u in the *Rating Matrix*R_{UI} according to the following rules.

—If the value of ($|Rev_{num}(u,i)^{pos}| - |Rev_{num}(u,i)^{neg}|$) is greater or equal than 0, then we add item i into the *Like+* part of the V_{UI} *(u)* with the value of +1.
—If the value of ($|Rev_{num}(u,i)^{pos}| - |Rev_{num}(u,i)^{neg}|$) is less than 0, then we add item i into the *Dislike-* parts of the V_{UI} *(u)* with the value of −1.
—If item $i$is unknown to user u, then we add item i into the *Unknown* parts of the V_{UI} *(u)* with the value of 0.

After we have built the virtual rating matrix, the standard user-based and item-based collaborative filtering algorithms can be utilized to predict item recommendations in different web applications.

6. EXPERIMENTS

6.1. Experimental Setup

6.1.1. Data and Tools. In order to validate the performance of our methods, we use two sets of data to conduct the experiments, which are respectively Youku dataset (which is without users' real ratings) and Amazon dataset (which is with real ratings).

To crawl the Youku datasets, we used nine Chinese queries to search videos.

{体育 ti-yu "sport", 音乐 yin-yu "music", 新闻 xin-wen "news", 科技 ke-ji "science", 旅游 lv-you "tourism", 电影 dian-ying "movie", 原创 yuan-chuang "originality", 汽车 qi-che "automobile", 时尚 shi-shang "fashion" }

Finally, we got the data[6] including more than 10,320 videos, each of which had more than 20 reviews. All the reviews were written in Chinese.

The Amazon datasets were crawled from the book section of the Amazon Web site in China, which includes more than 700000 reviews written in Chinese.

In Phase 1, a negation word list that contains ten Chinese negations was used.

{不 bu "not", 不会 bu-hui "would not", 没有 mei-you "don't have", 没 mei "don't have", 虽然 sui-ran "although", 虽 sui "although", 尽管 jin-guan "although", 缺 que "don't have", 缺乏 que-fa "don't have", 无 wu "don't have"}.

For all the experiments, the HowNet Sentiment Dictionary[7] was used as the sentiment dictionary, which is wellknown in the area of Chinese sentiment classification containing 4,566 positive words and 4,370 negative words.

There are more than 30 emotions provided by Youku for users to use while writing reviews. The following 10 positive emoticons and 5 negative emotions were used as the initial emoticon set and they were classified manually (−1 or +1). Then the *Updating step* (Section 4.1.4) is used to recalculate the strength of the sentiment polarity for each emotion.

{"smile", "love", "joking", "sweat", "naughty", "Uh-oh", "cool", "flower", "kiss", "thumbs up"}.

[6]http://learn.tsinghua.edu.cn:8080/2006990066/OVRdataset.html.
[7]http://www.keenage.com/download/sentiment.rar.

{“sad”, “sick up”, “angry”, “sweat”, “Tired”}.

6.1.2. The Collaborative Filtering Recommendation Algorithm. We use the standard user-based and item-based collaborative filtering algorithms conduct the recommendation experiments.

In user-based CF, to derive the recommendations for a target user u, k most-similar users are selected, which constitute the neighborhood of u, denote by *N(u)*. When predicting the rating of a given user u for an unknown item i, the rating score of i can be computed by:

$$r_{UI}(u,i) = \overline{R_{UI}(u)} + \frac{\sum_{v \in N(u)} w(u,v)(R_{UI}(v,i) - \overline{R_{UI}(v)})}{\sum_{v \in N(u)} w(u,v)}. \tag{10}$$

In the above equation, $R_{UI}(u,i)$ is rating value user u put on item i, the $\overline{R_{UI}(u)}$ is the mean rating for the user u and the weight $w(u,v)$ reflects the similarity between each user v and the given user u (i.e., the value of $S_{UI}(u,v)$). Then, the *Top N* items with the highest $r_{UI}(u,i)$ are selected in the recommendation list for the user u.

In the case of item-based CF, the prediction score is the average of the ratings on k most-similar items *N(i)* rated by the given user u. The prediction for a rating of a given user u for an item i is hence:

$$r_{UI}(u,i) = \frac{\sum_{j \in N(i)} w(i,j) R_{UI}(u,j)}{\sum_{j \in N(i)} w(i,j)}, \tag{11}$$

where the weight $w(i,j)$ reflects the similarity between each item j and the given item i (i.e., the value of $S_{UI}(i,j)$). Then, the *Top N* items with the highest $r_{UI}(u,i)$ are selected in the recommendation list for the user u.

As the rating matrixes are the input of the CF algorithms, we use *Virtual Rating*, *Real Rating* (if included in the datasets), and their averagevalue: *Real & Virtual Rating* to evaluate the effectiveness of the virtual rating.

6.2. Sentiment Classification Accuracy

We first tested the accuracy of our sentiment classification method by using a set of Youku[8] data with 1,085 videos and 6,450 users. Each video has at least 100 *video-oriented reviews*, and the total number of reviews is 120,174 in this set. Among the 120,174 reviews, there are 49,271 reviews that contain more than one emoticon. In the experiment, we set the value of parameter α in Equation (4) as default 0.3, because we considered that the emoticon sentiment score $RS^E(r)$ was more important than the word sentiment score $RS^W(r)$ for the sentiment classification.

After removing the noisy reviews as mentioned in Section 4, we manually labeled the polarities of 1000 reviews. The numbers of positive, end negative reviews in the labeled set are 653 and 347 respectively. We took the labeled data as the actual polarities of reviews.

We measured two approaches in the comparison, that is, SELC and SVM. In SELC, the method of Phase 1(unsupervised and semi-supervised models) proposed in this article was used to get the sentiment classification result. In *SVM*, the supervised Support Vector Machine classifier[9] was used to conduct the sentiment classification, and the SVM classifier with a linear kernel was ran in 10-fold stratified cross-validation mode. We used HowNet Sentiment Dictionary and an initial emoticon set as the feature set. Table I shows the sentiment classification's precision and recall results.

[8]Youku is YouTube counterpart in China.

[9]WEKA 3.4.11 was used (http://www.cs.waikato.ac.nz/ml/).

Table I. Result of the Sentiment Classification

Method	Review Sentiment	Precision	Recall	F1
SELC	Positive	92.8	96.4	94.6
	Negative	92.9	85.9	89.3
	Total	92.8	92.8	92.8
SVM	Positive	86.5	95.7	90.9
	Negative	89.9	71.8	79.8
	Total	87.4	87.4	87.4

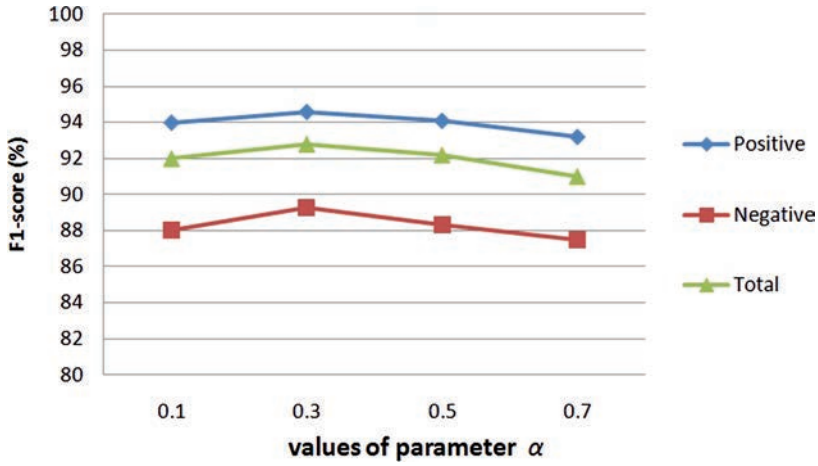

Fig. 4. Sentiment classification results for different values of parameter α.

From Table I, we can see that both the SELC achieves higher F_1 scores (92.8%) on *Total* reviews than the *SVM* classifier (87.4%). It is worth noting that SVM has suffered from the unbalance training data (pos:653, neg:347, the common ratio in real online environments) and gets bad recall values on negative reviews (71.8%). On the other side, *SELC* can still achieve a comparatively good recall on negative reviews (85.9%).

In Equation (4), the parameter $\alpha \in [0, 1]$ determines the weight put on each factor, that is , the balance weight between the review's word sentiment score $RS^W(r)$ and its emoticon sentiment score $RS^E(r)$. We have also designed an experiment for parameter sensitivity analysis.

In Equation (4) the default value of α if 0.3, we have set the value of α as 0.1, 0.3, 0.5 and 0.7 respectively. Figure 4 shows sentiment classification's F1-score results.

From Figure 4 we can see that the sentiment classification achieve the best result when $\alpha = 0.3$. When α is bigger than 0.3, the performance descends gradually alongwith the growing of α, which proves that hypothesis: the emoticon sentiment score $RS^E(r)$ is more important than the word sentiment score $RS^W(r)$ for the sentiment classification" is correct. Generally speaking, the performance is not sensitive to the parameter α, and the F_1-scores are close to each other considering different values of α.

There are several novel improvements in the *SECL* used in this article over the method of the original SELC model in Qiu et al. [2009], which affect the performance simultaneously. To check their individual effect, two variant models were implemented. They are referred to as *V1*and *V2* respectively. In *V1*, the new iteration control strategy is replaced by the iteration control method of the *original* SELC model. In *V2*, the emoticon analysis is removed from both unsupervised model and semi-supervised model.

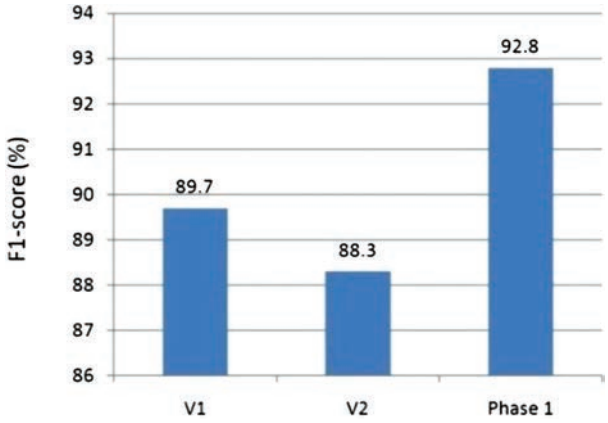

Fig. 5. The results of two variants of SELC in Phase 1 of the article.

Figure 5 shows that both the new iteration control strategy and the emoticon analysis have taken effect on the performance improvement, that is, improving 3.1% and 4.5% F_1-scores respectively. It suggests that the integration of emoticons can be very useful in further increasing the performance of the review sentiment classification, and the new iteration control strategy in the unsupervised model can also provide more accurate training data for the semi-supervised model.

Thus, the above analysis results indicate that our classification approach is capable of overcoming the challenges of online reviews' special features and providing reliable results for the building of virtual *Rating Matrixes* in the next phrase of producing item recommendations.

6.3. Item Recommendation Accuracy

6.3.1. Results of Recommendations on Youku Dataset. To compute recommendations, we classified 68,561 positive, 39,576 negative reviews on 1085 videos. The corresponding rating matrix was established for 6,450 users, with generated 61,137 virtual user-item ratings (the number of $+1$ and -1).

In our experiments, we compared the results for different approaches. Following is the description of labels we used to denote each of these algorithms.

—*YOUKU.* The recommendation approach of the Youku Web site, where each video is along with 3 recommended videos mainly based on video popularity.
—*User-SELC.* The User-based Collaborative Filtering Approach, where the results of *SELC* are used to predict the virtual ratings.
—*Item-SELC.* The Item-based Collaborative Filtering Approach, where the results of *SELC* are used to predict the virtual ratings.

The performance of video recommendations was then measured through statistical evaluation method.

Users are often split into training and test sets. The algorithm is trained over the users from the training set and evaluated over the users in the test set [Shani et al. 2008]. In this article, we evaluated the accuracy of recommendations using a "cold-start" protocol on the dataset. First, we randomly selected 860 (80%) of the items to be training items, leaving 217 (20%) as testing items. Then, we selected 500 users with the least item ratings to be test users.

Since each test user had rated two sets of items, that is, training item set and testing item set, we can evaluate the performance of our approach by calculating the precision

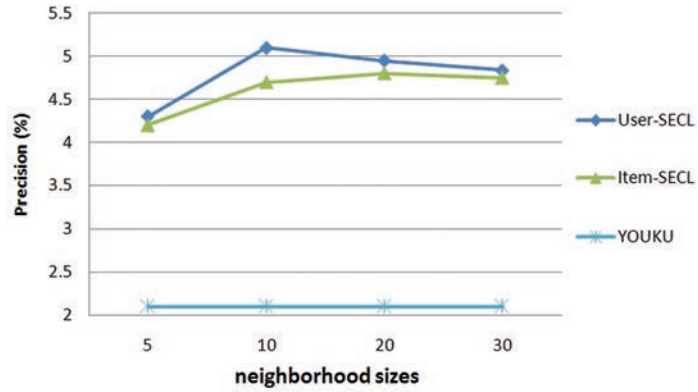

Fig. 6. The results of average precisions of Top 3 recommendations for *CF-based* approach.

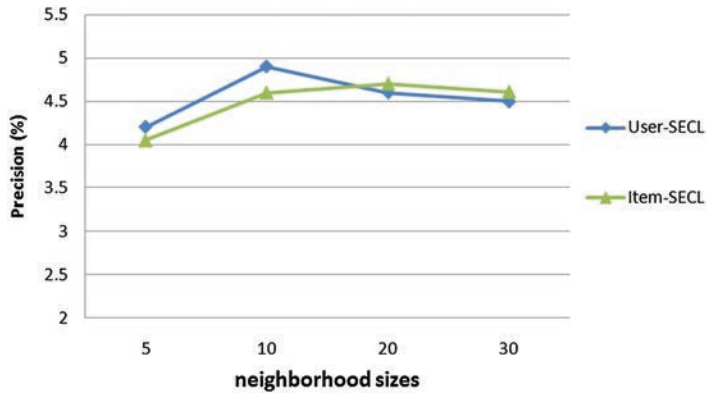

Fig. 7. The results of average precisions of Top 10 recommendations for CF-based approach.

of a fixed length of recommendation list [Gunawardana and Meek. 2009]. We first used the algorithm to derive a recommendation list based on the training items rated by a test user u. We then defined the per-user precision at the recommendation list containing *Top N* items as:

$$Precision(u) = \frac{HitNumber}{N}, \tag{12}$$

where *HitNumber* is the number of items in the recommendation list that are hit in the testing item set of user u. Then, we averaged the resulting per-user precisions over all the 500 test users to get an average precision of the *Top N* recommendation.

It's worth noting that in the process of statistical experimental simulation, since we can't get the real user ratings on the Youku dataset, we used the virtual ratings as the ground truth considering the high performance of the review sentiment analysis, which was validated in experiments of review sentiment classification with the F_1 scores of 92.8%. Additionally, we have also conducted a matching experiment to clarify the validity of using the virtual ratings as ground truth (please see Section 6.3.2).

Figure 6 and Figure 7 show the average precisions, respectively, of Top 3 and Top 10 recommendations for the baseline approaches with varying neighborhood sizes. Since YOUKU doesn't provide results for Top 10 recommendations, Figure 7 gives out only the results of the CF-based approaches.

In Figure 6 and Figure 7 we can see that both the CF-based approaches obviously outperform the YOUKU approach (2.1%). These two figures also show that the User-SELC approach achieves the best results for neighborhood size $k = 10$, which lead to the precisions of 5.1% at Top 3 recommendation and 4.9% at Top 10 recommendation, while the Item-SELC approach achieves its best results for $k = 20$, which lead to the precisions of 4.8% at Top 3 recommendation and 4.6% at Top 10 recommendation.

Given a test user u, user-based CF for Top N recommendation relies on similar users who have similar rating patterns. These users are more likely to rate the same test items as user u do. But item-based CF relies on items similar to the training items rated by user u. The test items of user u are not necessarily among the items similar to the training items rated by user u unless the test items are actually similar to the training items rated by the user (which is not always true). So, the results of item-based CF for Top N recommendation are generally poor compared to user-based CF.

Because the top-ranked videos are usually more noticeable to the online users, the precisions at Top 3 recommendations is more worthwhile to be noted in producing better recommendations. From the above two figures, we can see that the precisions at Top 3 recommendations set are also slightly better than those at Top 10 recommendations respectively by the Item-SELC and User-SELC approaches. In particular, User-SELC achieves the best result of 5.1% at Top 3 recommendations.

Experimental results show that the precision of the proposed approach is relatively low with best precision 5.1% (User-SELC at Top 3). That is mainly caused by the character of the date set we used. We have only about 18 (120,174/6,450) items rated for each user in average. So we have less than 4 items in the test set for a user in average. Because of the small size of the test set, the *HitNumber* in Equation (12) in the recommendation list is also very small, which causes the low precision.

There are other papers also encounter that problem. For example, in Gunawardana and Meek [2009], the precision of experiment on Ta-Feng dataset (with **23** items rated for each user in average) is also low (<4.5%), while the precision of experiment on *MovieLens* dataset (with **165** items rated for each user in average) is relatively high (best result 35%).

So in the case, we think the results are reasonable and we cannot say the system is not useful in reality only considering the low relatively precision.

6.3.2. Results of Recommendations on Amazon Dataset. Since we cannot get the real user ratings on the Youku Web site, we referred to the dataset of online books (Amazon China), which contains both the reviews and real ratings. The experiment is designed to measure the effectiveness of virtual rating compared to the real rating.

Like the experiments on Youku, the same statistical evaluation approach was used when we process the Amazon dataset; we got 318,730 reviews on 1,805 books and 28,254 reviews replied to other reviews. The number of users is 5502. Each of them has written about 5 reviews in average. The real rating made by user is an integer between 1 and 5, 5 means like the item very much, 1 means very dislike. After the procession of *Phase 1*, we also got the virtual rating set of the users. The virtual rating generated from the approach SELC in *Phase 1* is a fraction in [0, 5] (just a normalization of the review sentiment score).

First, in order to analyze the matching relationship between virtual rating and real rating, we added an experiment to see the proximity between those two kinds of ratings and clarify the effectiveness of the virtual ratings.

Given a user u and item i, we have the real rating u put on i: $r_{real}(u,i) \in \{1,2,3,4,5\}$ and its corresponding virtual rating u put on i: $r_{virtual}(u,i) \in [0\ 5]$. We built a virtual rating set $vrset_c$ for each real rating category $c, c \in \{1, 2, 3, 4, 5\}$. Considering a real rating category c, we get all the real ratings belong to it, and put all the virtual ratings corresponding

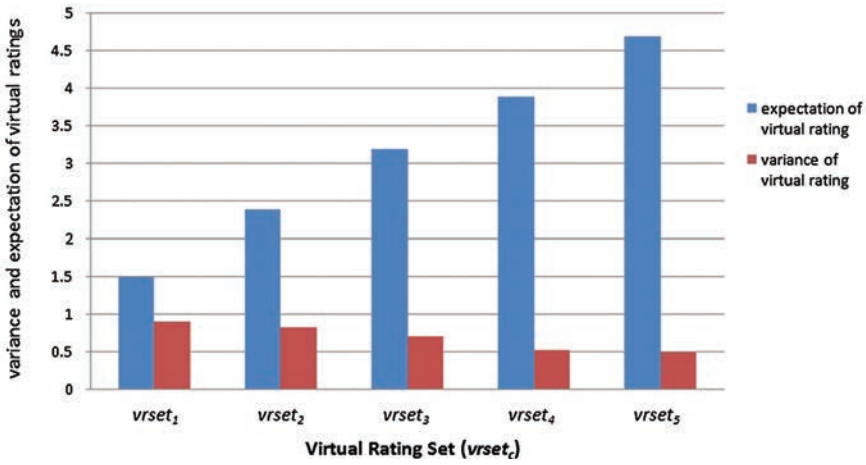

Fig. 8. The matching results of real and virtual ratings.

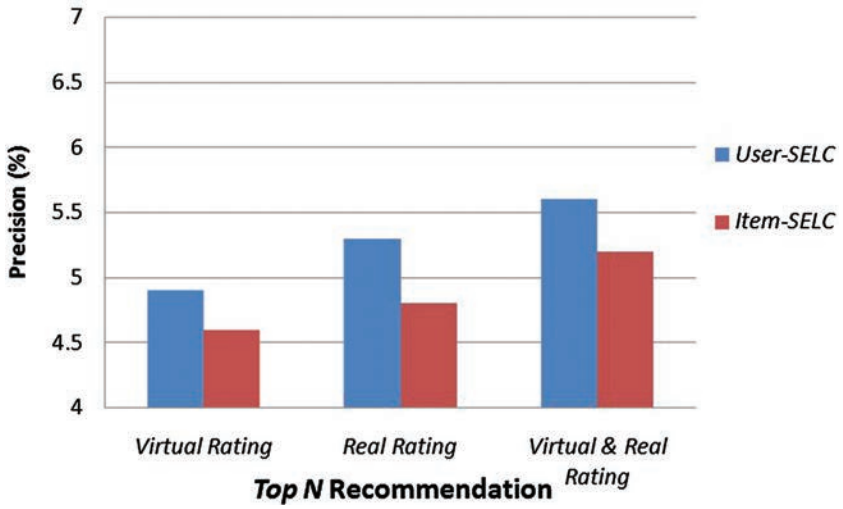

Fig. 9. The results of User-SELC and Item-SELC for Top 10 recommendations.

to these real ratings into the virtual rating set $vrset_c$. After the above steps, we have five virtual rating sets corresponding to the five real rating categories, that is, $vrset_1$, $vrset_2$, $vrset_3$, $vrset_4$, $vrset_5$.We evaluated the expectation and variance of the virtual ratings in each of the five virtual rating sets separately. Figure 8 shows the result.

From the results we can see that expectation in the each $vrset_c$ is close to its real rating category value, and the variances are acceptable. It suggests that the virtual ratings generated from sentiment analysis are efficient and make sense, and we can use virtual ratings as the ground truth for the dataset that does not contain real ratings.

Then, we used the CF-based approach (Section 6.1.2) to predict the recommendation on the dataset. According to the different type of item ratings, we designed three different kinds of subexperiments (using *Real Rating*, *Virtual Rating* and their averagevalue: *Real and Virtual Rating*), to evaluate the effectiveness of the virtual rating. For all the subexperiments, the user's real ratings are used as the ground truth. In

other word, we use the user's real ratings to evaluate the predicted results. Figure 9 shows the Top 10 recommendation results of the User-SELC (neighborhood size $k = 10$) and Item-SELC (neighborhood size $k = 20$) approaches.

From Figure 9 we can see that the User-SELC approach achieve better results than the Item-SELC approach considering all the three different kinds of ratings. What's more, the results of all the Virtual and Real Rating methods outperform the other two methods only using virtual rating or real rating. It suggests that the virtual ratings has addressed the rating sparsity limitation of current media-sharing sites to some extent and improved the applicability of collaborative filtering (CF) recommender techniques in these sites.

7. CONCLUSION AND FUTURE WORK

In this article, we developed review-aware recommender algorithms that particularly exploited the sentiment classification results to automatically derive virtual ratings, and then fused them into item-based and user-based CF algorithms by which the User-Item Rating Matrix can be inferred by decomposing item reviews that users gave to the items.

Through experiments on two datasets (one is without users' real ratings and another is with users' real ratings), we identified the significant impact of virtual ratings on augmenting recommenders. The results of the experiments show that 1) the SELC model achieves high precision on the review dataset and can produce virtual ratings of high quality; 2) the virtual ratings generated from the sentiment classification can be used to improve the recommender system on those resource sharing Web sites regardless of whether there are real ratings or not.

In the future, we will be engaged in classifying the reviews into more delicate categories in addition to "positive" and "negative" and exploring the application of the virtual ratings on the state-of-the-art recommendation algorithms. On the other hand, we will try to extend our method to other product domains, so as to additionally improve its cross-domain applicability.

REFERENCES

Alpaydin, E. 2004. *Introduction to Machine Learning*. MIT Press, Cambridge, MA.

Aue, A. and Gamon, M. 2005. Customizing sentiment classifiers to new domains: a case study. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*.

Balabanovic, M. 1998. Exploring versus exploiting when learning user models for text recommendation. *User Model. User-Adapt. Interact. 8*, 4, 71–102.

Blitzer, J., Drezde, M., and Pereira, F. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Breese, J. S., Heckerman, D., and C. Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI '98)*. 43–52.

Christakou, C. and Stafylopatis, A. 2005. A hybrid movie recommender system based on neural networks. In *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*.

Dave, K., Lawrence, S., and Pennock, D. M. 2003. Mining the Peanut gallery: opinion extraction and semantic classification of product documents. In *Proceedings of the International World Wide Web Conference*.

de Gemmis, M., Lops, P., Semeraro, G., and Basile, P. 2008. Integrating tags in a semantic content-based recommender. In *Proceedings of the ACM Conference on Recommender Systems (RecSys '08)*. 163–170.

Ganu, G., Elhadad, N., and Marian, A. 2009. Beyond the stars: Improving rating predictions using review text content. In *Proceedings of the 12th International Workshop on the Web and Databases*.

Gunawardana, A. and Meek, C. 2009. A unified approach to building hybrid recommender systems. In *Proceedings of the ACM Conference on Recommender Systems (RecSys'09)*. 117–124.

Jakob, N., Weber, S. H., Müller, M.-C., and Gurevych, I. 2009. Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion (TSA '09)*.

KI LEUNG, C. W., FAI CHAN, S. C., AND LAI CHUNG, F. 2006. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *Proceedings of the ECAI-Workshop on Recommender Systems*. 62–66.

KURUCZ, M., BENCZUR, A. A., AND CSALOANY, K. 2007. Methods for large scale SVD with missing values. In *Proceedings of KDD Cup Workshop at SIGKDD'07, 13th ACM International Conference on Knowledge Discovery and Data Mining*. 7–14.

NA, J. C., SUI, H., KHOO, C., CHAN, S., AND ZHOU, Y. 2004. Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. In *Proceedings of the 8th International ISKO Conference.* I.C. McIlwaine, Ed., 49–54.

PANG, B., LEE, L., AND VAITHYANATHAN, S. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

PAPAGELIS, M., PLEXOUSAKIS, D., AND KUTSURAS, T. 2005. Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences. In *Proceedings of the 3rd International Conference on Trust Management (iTrust 05)*. Springer, 224–239.

PATEREK, A. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of the KDD Cup Workshop at the 13th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD '07)*. 39–42.

QIU, L., ZHANG, W., HU, C., AND ZHAO, K. 2009. SELC: A self-supervised model for sentiment classification. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM '09)*. 929–936.

READ, J. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL-2005 Student Research Workshop*.

SALAKHUTDINOV, R. AND MNIH, A. 2008a. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*.

SALAKHUTDINOV, R. AND MNIH, A. 2008b. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, vol. 20.

SARWAR, B. M., KARYPIS, G., KONSTAN, J. A., AND RIEDL, J. 2000. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC '00)*. 285–295.

SETTEN, M. V. AND VEENSTRA, M. 2003. Prediction strategies in a TV recommender system: Method and experiments. In *Proceedings of the International World Wide Web Conference*.

SHANI, G., CHICKERING, M., AND MEEK, C. 2008. Mining recommendations from the Web. In *Proceedings of the ACM Conference on Recommender Systems (RecSys'08)*. 35–42.

TSO-SUTTER, K. H. L., MARINHO, L. B., AND SCHMIDT-THIEME, L. 2008. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the ACM Symposium on Applied Computing (SAC '08)*. 1995–1999.

TURNEY, P. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of documents. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics*.

YANG, B., MEI, T., HUA, X., YANG, L., YANG, S., AND LI, M. 2007. Online video recommendation based on multimodal fusion and relevance feedback. In *Proceedings of the International Conference on Image and Video Retrieval (CIVR '07)*. 73–80.

ZAGIBALOV, Z. AND CARROLL, J. 2008a. Unsupervised classification of sentiment and objectivity in Chinese text. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*. 304–311.

ZAGIBALOV, Z. AND CARROLL, J. 2008b. Automatic seed word selection for unsupervised sentiment classification of Chinese text. In *Proceedings of the 22nd International Conference on Computational Linguistics*. 1073–1080.

ZHANG, W., ZHAO, K., QIU, L, AND HU, C. 2009. SESS: A self-supervised and syntax-based method for sentiment classification. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information, and Computation*. 596–605.

Received August 2010; revised January 2011, May 2011, August 2011; accepted August 2011