Are Ratings Always Reliable? Discover Users' True Feelings with Textual Reviews

Bin Hao, Min Zhang(✉), Yunzhi Tan, Yiqun Liu, and Shaoping Ma

Department of Computer Science and Technology,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China
{haob15,tyz13}@mails.tsinghua.edu.cn, {z-m,yiqunliu,msp}@tsinghua.edu.cn

Abstract. In e-commerce systems, users' ratings play an important role in many scenarios such as reputation and trust mechanisms and recommender systems. A general assumption in these techniques is that users' ratings represent their true feelings. Although it has long been adopted in previous work, this assumption is not necessarily true.

In this paper, we first present an in-depth study of the inconsistency between users' ratings and their reviews. Then we propose an approach to mine users' "true ratings" which better represent their real feelings, from textual reviews based on Gated Recurrent Unit (GRU) and hierarchical attention techniques. One major contribution is that we are about the first, to the best of our knowledge, to investigate this new problem of discovering users' true ratings, and to provide direct solutions to revise ratings that are insincere and inconsistent.

Comparative experiments on a real e-commerce dataset have been conducted, which show that the "true ratings" learned by the proposed model is significantly better than the original ones in terms of consistency with the reviews in three sets of crowdsourcing-based evaluations. Furthermore, leveraging different state-of-art recommendation approaches based on the learned "true ratings", more effective results have been achieved at all times in rating prediction task.

Keywords: Rating revision · Review to score
Deep learning for recommendation

1 Introduction

In recommendation systems, users' ratings are widely used as a basis to learn user's preferences and make recommendations in most of the classical algorithms, such as [2]. In recent years, simultaneously exploiting ratings and reviews for recommendation attracts more and more attention for its ability to mitigate

This work is supported by Natural Science Foundation of China (Grant No. 61672311, 61532011).

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11108, pp. 442–453, 2018.
https://doi.org/10.1007/978-3-319-99495-6_37

data sparsity and build more accurate models [3,4]. In these models, however, ratings still play important roles in recommendation task or even been taken as the targets of learning.

For these ratings-based techniques, the fundamental assumption is that ratings are valid and reliable, which honestly indicate overall feelings of users towards items. While unfortunately, this assumption is not always true [1,5]. Apart from spam users which have been widely studied previously [6,7], a considerable number of ordinary users also give inconsistent ratings with the corresponding reviews in opinion expression. There have been some observations on this phenomenon [8] but little work has been done to overcome it, which is the major topic of this paper. Examples of ratings and their corresponding reviews given by real ordinary users, who have consumed the products or services, from a large-scale e-commerce website are shown in Table 1.

Table 1. Examples from real e-commerce dataset, where ratings fail to represent users' true feelings

No.	Ratings	Reviews
1	5.0	There was a hole on the socks. The user service wasn't very friendly
2	5.0	It was too large, especially the sleeves. The styles was Okay
3	5.0	Feel a bit hard, maybe because I just start to use
4	1.0	The clothes are of good quality, and look beautiful
5	3.0	Very beautiful! I like it. Praise!

In this paper, we first investigate the problem where ratings fail to represent users' opinion expressed in the reviews. Generally, the user's review shows more information on his experience with the item after he has consumed them, which takes more reliable information than a single rating in many cases. Therefore, a deep-learning based approach is then proposed for mining users' true feelings from textual reviews. Furthermore, we design three experiments for evaluation. For recommendation scenario, performances of the learned "true ratings" in rating prediction task, which has been examined via different classical recommendation algorithms. Compared with original ratings, significant improvements are achieved in all the experiments using the learned ratings.

The main contributions of this work are as follows: (1) To the best of our knowledge, it is about the first work on detecting biased user ratings and building models to directly mine users' "true ratings". (2) We show how deep-learning-based approach help in this new problem, representing an inherent connection between textual information and score ratings. Furthermore, various state-of-art recommendation algorithms achieved significant performance improvements by using the learned ratings. (3) The "true ratings" learned by our models can be applied as groundwork in many scenarios where user ratings are adopted.

2 Related Work

Although ratings in e-commerce systems are widely used in many scenarios as important feedback, they are not always reliable. Some sellers do encourage buyers to provide positive feedback and avoid negative feedback to show that consumers are satisfied [8]. The problem of spam users, wherein users promote or degrade targeted items intentionally through fraudulent ratings and reviews, is one of the reasons [7]. There has been a lot of work focusing on detecting spam users [9–11].

Moreover, a growing body of work in recent years has paid attention to simultaneously exploiting user ratings and reviews in order to improve the performance of recommendations [3,4,12]. In such algorithms, information from reviews is introduced to better model user preferences and item features. However, the assumption that the users' ratings are reliable still serves as a basic assumption and ratings are directly used in these approaches. Hence it is essentially different from the basic problem of this work, in which the ratings are supposed to be not necessarily reliable and are not encouraged to be used directly.

Differing markedly from previous work, our work focuses on the reliability of original ratings when they are used to represent the users' "true feelings" towards the items. This problem is crucial but has not been well studied. In this work, we propose a method to mine users' true feelings from their reviews in which they describe in detail their experiences of products and services after consuming them. To the best of our knowledge, this is about the first direct solutions that are proposed for this problem. We also show that properly revised ratings also help the previous recommendation algorithms which make use of rating information.

In general, reviews are written in a free text format with natural language. To mine users' "true ratings" accurately, we need to understand semantic and structural information in reviews, to which deep learning techniques are shown to be helpful. Recurrent neural networks (RNNs) are able to process arbitrary sequences of inputs and form some kind of short-term memory using their internal memories, which make them applicable to tasks such as speech recognition or text processing [13]. Theoretically, RNNs can efficiently represent more complex patterns. The Gated Recurrent Unit network (GRU) [14] is a special kind of RNNs and capable of learning long-term dependencies. Attention mechanism has already been shown effective in many areas, such as machine translation [15] and sentiment analysis [16]. [17] propose a hierarchical attention network for document classification. The main difference between us is that they calculate word attention to find the critical words in the sentence and we calculate it to find the critical words in the whole review.

3 Are Ratings Always Reliable?

As described above, ratings may sometimes fail to represent users' true feelings towards items (i.e., products, services and so on). To investigate this problem

with a real-world dataset, we collected a real dataset of clothing and accessories category from a popular e-commerce website. The dataset is a collection of feedback, where each piece of feedback consists of 4 factors: anonymized "UserID" and "ItemID", a numerical "rating" (from 1 to 5 stars) and a corresponding textual "review". In summary, there are 324,925 pieces of feedbacks provided by 284,848 users towards 27,370 items in the dataset (i.e. on average 11.8 feedback per item).

To measure users' true feelings towards items, we randomly selected 15,424 reviews (approximately 5% of the whole dataset) for crowdsourcing-based labeling. For each textual review, we hide its original rating and annotators were randomly selected to manually label five-level ratings according to the given corresponding reviews. Labeling quality was monitored in real time and unreliable annotators were interrupted in the task immediately by the crowdsourcing platform. Finally, 144 annotators provided valid labels and for each review, 3 labels were received. We also publish this dataset for the convenience of related researches.[1]

Some statistics of the labels are shown in Table 2. The average label variance is 0.2237, which indicates a good labeling consistency. We took the arithmetic average of the 3 labels as the review's labeled rating r^{lab}, which is used to represent user's true feelings towards an item in this work. It shows that the mean of all labeled ratings is 4.15, which is much lower than that of original ratings.

Table 2. Statistics of original and labeled ratings

Original mean	Labeled mean	Averaged labeled variance
4.46	4.15	0.2237

The distributions of the labeled and original ratings are shown in Fig. 1(a). 57.44% of original ratings are 5 stars, 35.78% are 4 stars, and less than 7% are 3 stars or less. If we treat 5 and 4 stars as positive feedback, more than 93% of feedback is positive. This percentage is similar to that is discovered in [8]. However, for labeled ratings, 40.06% of ratings are 5 stars and 41.52% are 4 stars, which shows about 12% less on positive ratings. Hence more diverse distribution is observed compared with the original one.

Another observation is, generally speaking, the differences between labeled ratings and original ones are mostly 1 or 2 levels. For example, the original rating of the review "The price of this hat is too high!" was 5; however, it's average label is 3.67. The original rating for the review "This looks good, and I like it very much." was 4, but was re-labeled as 5. Detail information of rating differences distribution is shown in Fig. 1(b). Here the rating difference is defined as the labeled ratings minus the corresponding original ratings. There are 33.85%

[1] It can be downloaded at https://pan.baidu.com/s/1O9r1S5ojGnrraivWwqT42w.

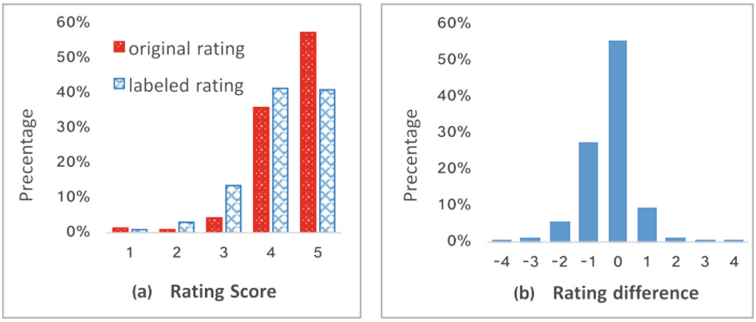

Fig. 1. Labeled ratings statistics. (a) the distribution on rating scores, (b) differences between ratings (labeled minus original). (Round labeled ratings to the nearest integer)

and 10.51% of reviews for which the labeled ratings are lower and higher than the original ratings, respectively, when we round the average labeled ratings to the nearest integer. If we compare the average labeled rating directly with the original one without round operation, then the proportions of the ratings by crowdsourcing labeling are respectively 46.62% lower than and 11.43% higher than the corresponding original ratings, which in total approaches to sixty percent of the data.

Table 3. Number of labeled ratings (columns, rounded to the nearest integer) v.s. that of original ratings (rows)

Ratings	1	2	3	4	5	Total
1	58	74	62	30	4	228
2	8	49	71	33	7	168
3	7	54	235	303	54	653
4	26	196	1206	3100	986	5514
5	28	125	650	2926	5132	8861
Total	127	498	2224	6392	6183	

Moreover, we analyze the distribution of labeled ratings relative to original ones, as shown in Table 3. What's interesting, from the table, some ratings express completely opposite opinions from what is described in the reviews. For example, the original rating of the review "The quality is really bad! I will never buy it again!" was 5; however, it was labeled as 1 by all the three annotators on account of the strong dissatisfaction it expressed. The original rating for the review "Good commodity. Its fabrics, colors and other aspects are also relatively satisfactory." was 1, which was re-labeled as 5 to reflect the user's satisfaction with the item. Similar examples can be found in ratings which are revised by 3

levels (e.g. 1 to 4, 5 to 2, etc.). It does happen in real scenarios when users make misunderstanding on the meaning of the rating stars.

4 Model for Mining Users' True Feelings from Reviews

To understand users' true feelings on the items, we propose to analyze the information expressed in textual reviews, from which we learn revised scores as the users' "true ratings". In general, reviews are written in a free text format. To mine users' "true ratings", we need to analyze the semantic and the structural information expressed in them, which is the strong point of deep learning techniques. As a result, we propose a **Hierarchical Total Attention (HTA)** model based on deep learning techniques to mine users' "true ratings" from their reviews.

4.1 Formalizations

We treat the review as a document d containing n sentences $\{S^1, S^2, \cdots, S^n\}$. The length of the k-th sentence S^k is l_k. The embeddings of the words in sentence S^k are $\{w_1^k, w_2^k, \cdots, w_{l_k}^k\}$.

4.2 Overview of HTA

The goal of our model is to predict a rating given its corresponding review. We treat each review as a document and use a hierarchical structure to capture the relation between sentences in one review and between words in the review. And we utilize the attention mechanism to automatically assign weights to each word and sentence. The structure of the HTA model is shown in Fig. 2. First we use word attention mechanism to get the vector representation of each sentence as $\{s_1, s_2, \cdots, s_n\}$. Then we use sentence attention mechanism to get the vector representation of the review as d. After this, we use a fully connected layer to get the prediction value of the review r.

4.3 From Word to Sentence Vector

The embeddings of the words in each sentence are inputted and processed by bi-directional Gated Recurrent Unit(GRU)[14]. The k-th word embedding of the i-th sentence w_k^i is encoded as h_k^i. Then we use the attention mechanism to calculate the attention value of **the total words** in the review as follows:

$$score(h_i^j) = v_w^{\mathrm{T}} tanh(W_w h_i^j + b_w) \tag{1}$$

$$a_i^j = \frac{exp(score(h_i^j))}{\sum_{j=1}^{n} \sum_{k=1}^{l_j} exp(score(h_k^j))} \tag{2}$$

$$s_i = \sum_{j=1}^{l_i} a_i^j h_i^j \tag{3}$$

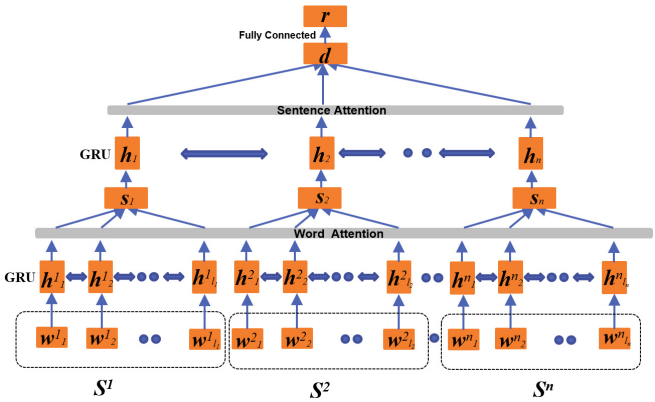

Fig. 2. The neural network architecture of HTA model.

First we calculate each word's importance of the review in Eq. 1, where score(h_i^j) is a score function which scores the importance of each word in the review, v_w is a word level context vector and v_w^{T} denotes its transpose, W_w is the weight matrix, b_w is the bias. The next step we use softmax function to calculate the attention weight of each word a_i^j in Eq. 2, which is the main difference from [17]. Then we aggregate all the word encoded vector h_i^j of a sentence to get its vector representation in Eq. 3.

4.4 From Sentence to Review Vector

We input each sentence vector to bi-directional Gated Recurrent Unit(GRU) encoder. The i-th sentence vector of review s_i is encoded as h_i. Then we use the attention mechanism similar to the word attention to select critical sentences to form the document representation. The document vector representation is formed via:

$$d = \sum_{i=1}^{n} a_i h_i \tag{4}$$

where a_i is the attention weight of sentence's encoded GRU vector h_i, which can be calculated similar to the word attention.

4.5 Regression and Learning

As d is extracted from words and sentences from the review, it can be treated as the feature vector of the review. We use a fully connected layer and a non-linear transformation (Relu) to get the final rating r of the review:

$$r = Relu(W_d d + b_d) \tag{5}$$

In this model, all of the parameters are learnt by minimizing the sum of squared errors between ratings labeled manually and ratings mined from reviews, which is shown as follows.

$$L(R) = \sum_{r^{lab} \in R} (r - r^{lab})^2 + \lambda_1 \|W_d\|^2 + \lambda_2 \|b_d\|^2 \tag{6}$$

where R denotes the training set of the labeled rating dataset, $\|\cdot\|$ denotes the l_2-norm. And the component $\lambda_1 \|W_d\|^2 + \lambda_2 \|b_d\|^2$ is used for regularization to avoid over-fitting. We also use a *dropout* technique to avoid over-fitting.

5 Experiment and Discussion

5.1 Dataset and Experimental Settings

As described in Sect. 3, we collected a dataset D of clothing and accessories from a popular e-commerce website. Then 15,424 feedbacks are randomly selected L from D which annotators from crowdsourcing platform manually labeled their ratings according to the feelings users expressed in the textual reviews.

We treated the labeled ratings from the reviews as the representation of users' true feelings towards items, i.e., the "true ratings", and designed HTA model to learn the connection between the textual reviews and the "true ratings". The labeled dataset is randomly divided the labeled dataset L into a training set R, a validation set V, and a testing set T. Specifically, 70% of the labeled dataset was used for training, 10% for validation and the remaining 20% for testing. We use the word embeddings from Google trained from Word2Vector model, each word is presented as a 200 dimension vector. We use five models as our baselines: Linear Regression (LR), SVM with linear kernel (LinearSVR), Multi-layer Perceptron with one hidden layer of 100 nodes (MLP), SVM with RBF kernel (SVR), HAN model depicted in [17]. For the top four models, we use the average vector of all words in a review as its feature and for the HAN model, we use the same setting as our model.

To evaluate the performance of our models, we adopted the commonly used metrics root mean squared error (RMSE) and mean absolute error (MAE), which are defined as below:

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (f_j - y_j)^2}, MAE = \frac{1}{n} \sum_{j=1}^{n} |f_j - y_j| \tag{7}$$

where f_j and y_j denote the prediction value and the true value, respectively.

5.2 Effectiveness of "True Ratings"

Evaluation of Rating Score Revision. This evaluation task is to measure the effectiveness of direct rating score revision by estimating the proper rating given a piece of related review. Crowdsourcing labeled "true ratings" are taken as the ground truth. Performances on RMSE on the testing set T by HTA model are given in Table 4. Here, the "Ori" denotes the original ratings and the "HTA" denote the ratings learned by HTA model, etc.

Table 4. Performance of HTA Model

Metric	Ori	SVR	LR	LinearSVR	MLP	HAN	HTA
RMSE	0.8163	0.6650	0.5936	0.5879	0.5711	0.4836	**0.4803**
MAE	0.5393	0.5078	0.4567	0.4323	0.4291	0.3497	**0.3431**

Evaluation with Pairwise Preference. To make a quantitative evaluation, we conducted experiments of pairwise preference to examine the consistency of the learned ratings with the feelings expressed in the reviews by crowdsourcing-based labeling. In detail, we randomly select 1,000 textual reviews. For easily distinguishing, we only selected the reviews of which the ratings learned by the HTA model are at least 1 level different from their original ratings. We then rounded the rating to the nearest integer to make the two types of ratings have the same appearances, hence no bias was introduced to annotators. Finally, for each review, 5 randomly selected annotators were asked to discern which rating was more consistent with the feeling expressed by the review. If an annotator found that the learned rating was more consistent, we added an "R2S" tag to the review; otherwise, an "Ori" tag was added. Then we calculated the percentages of reviews with different numbers of "R2S" labels, and the results are shown in Table 5. We can see that almost half of the scores from our model get 5 "R2S" and 86.9% scores of our model better than the original ones.

Table 5. The percentages of reviews with different numbers of "R2S" labels

	#"R2S" = 5	#"R2S" >= 3	#"R2S" < 3	#"R2S" = 0
HTA	**48.8%**	**86.9%**	13.1%	3.2%

Evaluation with Preferences on Recommend-Ability. From the above two sets of experiments, it is shown that given the users' review information, ratings from the proposed model work better than the original ones in terms of theirs consistencies to the users' feelings that are expressed in reviews. But there might be some other concerns: it is possible that a user does want to rate the item with the exact score he gives, while his comments are incomplete in expressing his opinion to the item. Here we call it review bias.

We design the third experiments that measure the quality of the revised ratings by crowdsourcing platform and try to reduce review bias. For this time the URLs of the items in the e-commerce system are shown to the crowdsourcing annotators directly. They are asked to click the link and browse the full information of the item, including the item descriptions and the corresponding complete reviews and ratings. Then the annotators are asked to label the recommend-ability, i.e. "whether the item is worthy of being recommended" in 5 levels, say "must not be recommended", "not a good candidate", "just so-so", "good recommendation candidate" and "strongly recommended". Each item is evaluated

by three annotators. Data is removed if the labels are not reliable, for example, the annotation procedure is too short to give a careful evaluation.

Finally, 978 items are annotated, and in total 2,459 (rating, review) pairs of these items are found in the previous experimental dataset. Taking the average score of the crowdsourcing labels on items recommend-ability as the ground truth, we evaluate the gap between the average user rating and the recommend-ability of an item with RMSE. Three types of ratings are measured: the original ratings (noted as "Original"), HTA model revised ratings ("R2S-HTA"), and the manually revised ratings according to the reviews by crowdsourcing experiments.

The Results are shown in Table 6. It verifies revised ratings achieved by the proposed models are significantly better on indicating whether the item is worthy of being recommended. Encouragingly, the performances of the HTA model is similar to that of the manually revised scores by the crowds.

Table 6. The gap between items recommend-ability and the ratings

	Original	R2S-HTA	Manual
RMSE	1.5294	1.2317**	1.2110**

**$p < 0.01$ compared with the Original one.

5.3 Effectiveness for Rating Prediction

Rating prediction is an important research topic in the field of personalized recommendation. The aim of rating prediction is to estimate the rating score a user, say u, will rate for any item i. The rating prediction task differs greatly from the previous rating score revision task because the textual reviews are not available and historical information is taken into consideration here.

We evaluate the effectiveness of ratings learned by our model in rating prediction task with two of the state-of-art rating prediction methods, i.e., MF and BMF, which have been popularly used in recommendation tasks in recent years.

The matrix factorization (MF) model [2]: this model maps user preference distributions q_u and item feature distributions p_i to a joint latent factor space of dimensionality f. It estimates the rating a user u will give to any item i by using $\hat{r}_{u,i} = q_u^T p_i$.

The biased matrix factorization (BMF) model [18]: differing from the MF model, the BMF model explains the full rating value not only by the interaction of users and items but also by the biases associated with either users or items. It predicts the rating of item i by user u by using $\hat{r}_{u,i} = \mu + b_u + b_i + q_u^T p_i$, where u, b_u, b_i denote the global average, user bias and item bias, respectively.

The MF and BMF models exploit the past ratings for training, and there are two kinds of ratings: original ratings ("Ori"), ratings learned by the HTA model ("HTA"). Whether using the MF or BMF models, we keep all settings the same except for the sources of ratings. Specifically, we use ratings associated with the user-item pairs in the whole dataset D except the test set in the labeled dataset T for training and try to predict the ratings of the user-item pairs in dataset T.

The crowdsourcing labeled ratings of dataset T are treated as the ground truth. For fair comparison, the information of dataset T is never used in any training procedure.

Table 7 is shown that relative to different rating prediction methods, we are able to achieve better performance by adopting the ratings learned from textual reviews. The result verifies our idea that ratings learned from reviews are more consistent with users' true feelings; hence we can model users' preferences more accurately based on them compare with the original ratings.

Table 7. Performance of rating prediction

Rating	MF		BMF	
	RMSE	MAE	RMSE	MAE
Ori	0.7470	0.5329	0.7339	0.5560
SVR	0.7185	0.5701	0.7242	0.5813
LR	0.6377	0.4959	0.6498	0.5202
LinearSVR	0.6622	0.5023	0.6078	0.4837
MLP	0.6198	0.4745	0.6304	0.5045
HAN	0.6021	0.4611	0.6036	0.4792
HTA	**0.5691**	**0.4326**	**0.5807**	**0.4582**

6 Conclusions and Futurework

In this work, we investigate the basic problem where users' ratings fail to represent their true feelings. We conduct extensive empirical analysis on real-world dataset and propose a deep-learning based approach to recover users' "true ratings" from textual reviews. The ratings learned by our models are able to provide a more reliable basis for the rating-based tasks. Experimental results on "true rating" revision task show that the learned ratings are more consistent with the reviews compared with the original ratings, and the performance of rating prediction task has also been improved by adopting the learned ratings.

The main contributions of this work are: (1) To the best of our knowledge, this is about the first work to detect the unreliability of user ratings and build models so as to recover users' "true ratings"; (2) We show the power of deep learning approach to learn users' "true ratings" from their reviews, which reveals the inherent connection between textual information and score ratings; (3) Our learned "true ratings" also help significantly on rating prediction task. This work can be taken as the foundation in varied scenarios.

References

1. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decis. Support Syst. **43**(2), 618–644 (2007)
2. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **8**, 30–37 (2009)
3. Almahairi, A., Kastner, K., Cho, K., Courville, A.: Learning distributed representations from reviews for collaborative filtering. In: Recsys, pp. 147–154 (2015)
4. Bao, Y., Fang, H., Zhang, J.: Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In: 28th AAAI Conference, pp. 2–8 (2014)
5. Zhang, Y., Zhang, H., Zhang, M., Liu, Y., Ma, S.: Do users rate or review? Boost phrase-level sentiment labeling with review-level sentiment classification. In: SIGIR, pp. 1027–1030 (2014)
6. Gunes, I., Kaleli, C., Bilge, A., Polat, H.: Shilling attacks against recommender systems: a comprehensive survey. Computer **42**(4), 767–799 (2014)
7. Zhang, Y., Tan, Y., Zhang, M., Liu, Y., Ma, S.: Catch the black sheep: unified framework for shilling attack detection based on fraudulent action propagation. In: IJCAI, pp. 2408–2414 (2015)
8. Resnick, P., Zeckhauser, R.: Trust among strangers in internet transactions: empirical analysis of eBay's reputation system. the economics of the internet and e-commerce. Adv. Appl. Microeconomics **11**, 127–157 (2002)
9. Bhaumik, R., Mobasher, B., Burke, R.D.: A clustering approach to unsupervised attack detection in collaborative recommender systems. In: ICDM, pp. 181–187 (2011)
10. Burke, R., Mobasher, B., Williams, C., Bhaumik, R.: Classification features for attack detection in collaborative recommender systems. In: KDD, pp. 542–547 (2006)
11. Hurley, N., Cheng, Z., Zhang, M.: Statistical attack detection. In: Recsys, pp. 149–156 (2009)
12. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: 7th ACM Conference on Recommender Systems, pp. 165–172. ACM, Hong Kong (2013)
13. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT press, Cambridge (2016)
14. Chung, J., Gulcehre, C., Cho, K.H., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
15. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
16. Chen, H., Sun, M., Tu, C., Lin, Y., Liu, Z.: Neural sentiment classification with user and product attention. In: EMNLP, pp. 1650–1659 (2016)
17. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489 (2016)
18. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. In: Advances in Artificial Intelligence, vol. 4 (2009)