A Normal-Distribution Based Reputation Model

Ahmad Abdel-Hafez[1], Yue Xu[1] and Audun Jøsang[2]

[1] Queensland University of Technology, Brisbane, Australia
{a.abdelhafez, yue.xu}@qut.edu.au
[2] University of Oslo, Oslo, Norway
josang@mn.uio.no

Abstract. Rating systems are used by many websites, which allow customers to rate available items according to their own experience. Subsequently, reputation models are used to aggregate available ratings in order to generate reputation scores for items. A problem with current reputation models is that they provide solutions to enhance accuracy of sparse datasets not thinking of their models performance over dense datasets. In this paper, we propose a novel reputation model to generate more accurate reputation scores for items using any dataset; whether it is dense or sparse. Our proposed model is described as a weighted average method, where the weights are generated using the normal distribution. Experiments show promising results for the proposed model over state-of-the-art ones on sparse and dense datasets.

Keywords: Reputation Model, Ratings Aggregation, Uncertainty.

1 Introduction

People are increasingly dependent on information online in order to decide whether to trust a specific object or not. Therefore, reputation systems are an essential part of any e-commerce or product reviews websites, where they provide methods for collecting and aggregating users' ratings in order to calculate the overall reputation scores for products, users, or services [1]. The existence of reputation scores in these websites helps people in making decisions about whether to buy a product, or to use a service, etc. Reputation systems play a significant role in users' decision making process.

Many of the existing reputation models did not mention how good they are with different sparsity datasets, Lauw et.al [2] mentioned that the simple average method would be adequate with dense dataset supported by the law of large numbers [3]. Other models focused on robustness of the reputation score, i.e., the value is not easy to be affected by malicious reviews [4]. In general, the majority of the recently proposed reputation systems involved other factors, besides the ratings, such as the time when the rating was given or the reputation of the user who gave that rating. Usually, this data is incorporated with ratings as weights during the aggregation process, performing the weighted average method. These factors can be easily combined into our proposed methods.

11th International Conference on Trust, Privacy & Security in Digital Business (TrustBus 2014)
Munich, September 2014.

One of the challenges that face any reputation model is its ability to work with different datasets, sparse or dense ones. Within any dataset some items may have rich rating data, while others, especially new ones, have low number of ratings. Sparse datasets are the ones that contain higher percentage of items which do not have many ratings or users who didn't rate many items. However, with the increased popularity of rating systems on the web particularly, sparse datasets become denser by time as ratings build up on the dataset. Most of the current reputation models did not mentioned if they work well with dense or sparse datasets or both, others focused on sparse dataset only assuming they are the ones require attention only [2].

On the other hand, most of the existing reputation models don't consider the distribution of ratings for an item, which should influence its reputation. In this paper, we propose to consider the frequency of ratings in the rating aggregation process in order to generate reputation scores. The purpose is to enhance accuracy of reputation scores using any dataset no matter whether it is dense or sparse. The proposed methods are weighted average methods, where the weights are assumed to reflect the distribution of ratings in the overall score. An important contribution of this paper is a method to generate the weights based on the normal distribution of the ratings. We evaluate the accuracy of our results using ratings prediction system, and we compare with state-of-the-art methods. Our methods show promising results dealing with any dataset no matter whether it is dense or sparse.

In the rest of this paper, we will first introduce existing product reputation models briefly in Section 2, and then we will explain the proposed methods in Sections 3. We will also provide detailed experiments and results evaluation in Section 4 in order to prove the significance of our proposed method. Finally in Section 5 we conclude the paper.

2 Related Works

Reputation systems are used with many objects, such as webpages, products, services, users, and also in peer-to-peer networks, where they reflect what is generally said or believed about the target object [5]. Item's reputation is calculated based on ratings given by many users using a specific aggregation method. Many methods used weighted average as an aggregator for the ratings, where the weight can represent user's reputation, time when the rating was given, or the distance between the current reputation score and the received rating. Shapiro [6] proved that time is important in calculating reputation scores; hence, the time decay factor has been widely used in reputation systems [6–9]. For example, Leberknight et al. [9] discussed the volatility of online ratings, where the authors aimed to reflect the current trend of users' ratings. They used weighted average where old ratings have less weight than current ones. On the other hand, Riggs and Wilensky [10] performed collaborative quality filtering, based on the principle of finding the most reliable users. One of the baseline methods we use in this paper is proposed by Lauw et al., which is called the Leniency-Aware

Quality (LQ) Model [2]. This model is a weighted average model that uses users' ratings tendency as weights. The authors classified users into lenient or strict users based on the leniency value which is used as a weight for the user's ratings.

Another baseline model that we use was introduced by Jøsang and Haller, which is a multinomial Bayesian probability distribution reputation system based on Dirichlet probability distribution [8]. This model is probably the most relevant method to our proposed method because this method also takes into consideration the count of ratings. The model introduced in [8] is a generalization to their previously introduced binomial Beta reputation system [11]. The authors indicated that Bayesian reputation systems provide a statistically sound basis for computing reputation scores. This model provides more accurate reputation values when the number of ratings per item is small because the uncertainty in these cases is high. Using fuzzy models are also popular in calculating reputation scores because fuzzy logic provides rules for reasoning with fuzzy measures, such as trustworthy, which are usually used to describe reputation. Sabater & Sierra proposed REGRET reputation system [12] , which defines a reputation measure that takes into account the individual dimension, the social dimension and the ontological dimension. Bharadwaj and Al-Shamri [13] proposed a fuzzy computational model for trust and reputation. According to them, the reputation of a user is defined as the accuracy of his prediction to other user's ratings towards different items. Authors also introduced reliability metric, which represent how reliable is the computed score.

In general, some of the proposed reputation systems compute reputation scores based on the reputation of the user or reviewer, or they normalize the ratings by the behavior of the reviewer. Other works suggested adding volatility features to ratings. According to our knowledge, most of the currently used aggregating methods in the reputation systems do not reflect the distribution of ratings towards an object. Besides, there are no general methods that are robust with any dataset and always generate accurate results no matter whether the dataset is dense or sparse, for example, LQ model [2] is good with sparse datasets only and Jøsang and Haller model [8] generates more accurate reputation scores for items with low frequent ratings.

3 Normal Distribution Based Reputation Model (NDR)

In this section we will introduce a new aggregation method to generate product reputation scores. Before we start explaining the method in details, we want to present some definitions. First of all, in this paper we use arithmetic mean method as the Naïve method. Secondly, the term "rating levels" is used to represent the number of possible rating values that can be assigned to a specific item by a user. For example, considering the well-known five stars rating system with possible rating values of $\{1, 2, 3, 4, 5\}$, we say that we have five rating levels; one for each possible rating value.

As mentioned previously, the weighted average is the most currently used method for ratings aggregation, while the weights usually represent the time when the rating was given, or the reviewer reputation. In the simplest case, where we don't consider other factors such as time and user credibility, the weight for each rating is $\frac{1}{n}$, if there are n ratings to an item. No matter for the simplest average method or the weighted average methods that take time or other user related factors into consideration, the frequency of each rating level is not explicitly considered. For example, assume that an item receives a set of ratings $< 2, 2, 2, 2, 3, 5, 5 >$, for the simplest average method, the weight for each of the ratings is $\frac{1}{7}$ even the rating level 2 has higher frequency than the other two rating levels. For other weighted average methods, the weights are only related to time or some user related factors but not rating frequency.

In the following discussion, we will use the Naïve method as an example to explain the strength of our proposed method since the other factors can be easily combined into our methods to make the weights related to other factors such as time or user credibility.

Our initial intuition is that rating weights should relate to the frequency of rating levels, because the frequency represents the popularity of users' opinions towards an item. Another important fact that we would like to take into consideration in deriving the rating weights is the distribution of ratings. Not losing generality, like many "natural" phenomena, we can assume that the ratings fall in normal distribution. Usually the middle rating levels such as 3 in a rating scale $[1 - 5]$ system is the most frequent rating level (we call these rating levels "Popular Rating Levels") and 1 and 5 are the least frequent levels (we call these levels "Rare Rating Levels"). By taking both the rating frequency and the normal distribution into consideration, we propose to 'award' higher frequent rating levels, especially popular rating levels, and 'punish' lower frequent rating levels, especially rare rating levels.

Table 1: Comparing weights of each rating level between Naïve and NDR methods.

Ratings	**Rating Weight**		**Rating Weight**	
	Naïve	**NDR**	**Naïve**	**NDR**
2	0.1429	0.0765	0.5714	0.604
2	0.1429	0.1334		
2	0.1429	0.1861		
2	0.1429	0.208		
3	0.1429	0.1861	0.1429	0.1861
5	0.1429	0.1334	0.2857	0.2099
5	0.1429	0.0765		

Table 1 shows the difference between the Naïve method and the proposed Normal Distribution based Reputation Model (NDR) which will be discussed in

Section 3.1. From the second column in Table 1 (i.e., Weight per rating), we can notice that using the Naïve method the weight for each rating is fixed which is $\frac{1}{7} = 0.1429$. Different from the Naïve method, the NDR method generates different weights for different ratings, especially, the weights from rare ratings such as 2 and 5 to popular ratings such as 3 are increase and the increment is non-linear. This non-linear increase in weights for repeated ratings of the same level will result in a higher aggregated weight for that rating level. For example, rating level 2 is the most frequent level, in comparison, the aggregated weight generated by the Naïve method for rating level 2 is 0.5714, where the NDR model generates a higher value 0.604 which reflects the contribution from the frequency of rating level 2. On the other hand, rating level 3 gets a higher weight 0.186 in the NDR method than the Naïve method which generates a weight value 0.1429, however, this is not because level 3 is more frequent, but because it is a popular rating level. In contrast, rating Level 5 gets a lower weight in the NDR method because it is a rare rating level and not very frequent in this example.

3.1 Weighting Based on a Normal Distribution

Our method can be described as weighted average where the weights are generated based on both rating distribution and rating frequency. As mentioned above, we use a normal distribution because it represents many "natural" phenomena. In our case, it will provide different weights for ratings, where the more frequent the rating level is, the higher the weight the level will get. In other words, using this weighting method we can assign higher weights to the highly repeated ratings, which we believe will reflect more accurate reputation tendency.

Suppose that we have n ratings for a specific product P, represented as a vector $R_P = \{r_0, r_1, r_2, \dots, r_{n-1}\}$ where r_0 is the smallest rating and r_n is the largest rating, i.e., $r_0 \leq r_1 \leq r_2 \leq \dots \leq r_{n-1}$. In order to aggregate the ratings, we need to compute the associated weights with each rating, which is also represented as a vector $W_P = \{w_0, w_1, w_2, \dots, w_{n-1}\}$. As we discussed previously, the weights to the ratings will be calculated using the normal distribution density function given in Equation 1, where a_i is the weight for the rating at index $i, i = 0, \dots, n-1$, μ is the mean, σ is the standard deviation, and x_i is supposed to be the value at index i; the basic idea is to evenly deploy the values between 1 and k for the rating scale $[1, k]$ over the indexes from 0 to $n-1$. k is the number of levels in the rating system, in this paper we use the popular 5-star system, then $k = 5$.

$$a_i = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \tag{1}$$

$$x_i = \frac{(k-1) \times i}{n-1} + 1 \tag{2}$$

Equation 2 is used to evenly deploy the values of x_i between 1 and k, where $x_0 = 1$ and $x_{n-1} = k$. In Equation 1, the value of the mean is fixed, i.e., $\mu = \frac{(k+1)}{2}$. However, the value of is the actual standard deviation value extracted from the ratings to this item; hence, each item in the dataset will have different flatness for its normal distribution curve.

The purpose of using such these values for x, μ and σ is to produce normally distributed weights associated with the k-levels rating system. The generated weights in Equation 1 is then normalized so the summation of all weights is equal to 1, hence, we create the normalized weights vector $W_P = \{w_0, w_1, w_2, \ldots, w_{n-1}\}$ using Equation 3.

$$w_i = \frac{a_i}{\sum_{j=0}^{n-1} a_j}, \text{ where } \sum_{i=0}^{n-1} w_i = 1 \tag{3}$$

In order to calculate the final reputation score, which is affected by the ratings and the weights, we need to sum the weights of each level separately. To this end, we partition all ratings into groups based on levels, $R^l = \{r_0^l, r_1^l, r_2^l, \ldots, r_{|R^l|-1}^l\}, l = 1, 2, \ldots, k$, for each rating $r \in R^l$, $r = l$. The set of all ratings to item p is $R_P = \bigcup_{l=1}^{k} R^l$. The corresponding weights for the ratings in R^l are represented as $W^l = \{w_0^l, w_1^l, w_2^l, \ldots, w_{|R^l|-1}^l\}$

The final reputation score is calculated as weighted average for each rating level using Equation 4, where LW^l is called level weight which is calculated in Equation 5

$$NDR_p = \sum_{l=1}^{k} \left(l \times LW^l\right) \tag{4}$$

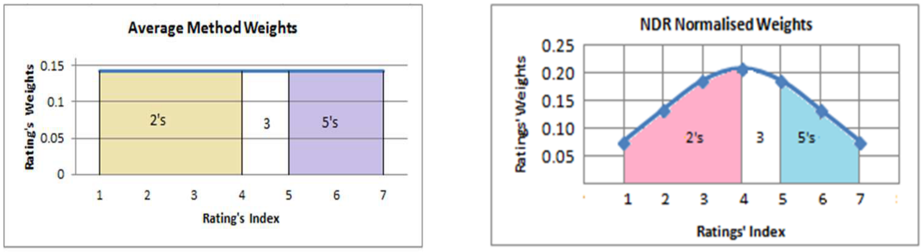
$$LW^l = \sum_{j=0}^{|R^l|-1} w_j^l \tag{5}$$

Equation 5 calculates level weights LW^l as a summation of the weights of every rating belonging to that level.

Fig. 1 shows the weights generated for the above example by the Naïve method and the proposed NDR method, where left-most region represents the overall weight for rating level 2, and the middle region and the right-most region are for rating levels 3 and 5. We can see that, the weights for all ratings are the same in Fig. 1a, which uses the Naïve method, while using the NDR method in Fig. 1b, the ratings with index near to the middle will be given higher weights.

3.2 Enhanced NDR Model by Adding Uncertainty (NDRU)

In this section we will do a modification to our proposed NDR method by combining uncertainty principle, introduced by Jøsang and Haller Dirichlet method [8]. This enhancement is important to deal with sparse dataset, because when the number of ratings is small, the uncertainty is high. The enhanced method is expected to pick up the advantages of both reputation models, i.e., the NDR method and the Dirichlet method. Inspired by the Dirichlet method in [8], the NDRU reputation score is calculated using Equation 6 which takes uncertainty



(a) Average method weights for the 7 ratings example.

(b) NDR normalised weights for the 7 ratings example.

Fig. 1: Weights generated using Naïve and NDR methods

into consideration:

$$NDRU_{p1} = \sum_{l=1}^{k} \left(l \times \left(\frac{n \times LW^{l} + C \times b}{C + n} \right) \right) \tag{6}$$

C is a priori constant which is set to 2 in our experiments, and $b = \frac{1}{k}$ is a base rate for any of the k rating values.

The NDRU method will reduce the effect of praising popular rating levels and depreciating rare rating levels process done by the NDR model. We can say that in all cases if the NDR method provides higher reputation scores than the Naïve method, then the NDRU method will also provide higher reputation scores but marginally less than the NDR ones and vice versa. However, as we have mentioned before, in the case of having a small number of ratings per item, the uncertainty will be higher because the base rate b is divided by the number of ratings plus a priori constant $n + C$ in Equation 6. In this case, the difference between the final reputation scores of the NDR and NDRU methods is noticeable. This advantage of the Dirichlet method to deal with sparse data is adopted by the NDRU method. Yet, when we use dense dataset, the difference between the final reputation scores of the NDR and NDRU methods will be very small, which allow the NDRU to behave similarly to the NDR method.

4 Experiment

In the beginning we want to say that there are no globally acknowledged evaluation methods that appraise the accuracy of reputation models. However, we choose to assess the proposed model in regards to the accuracy of the generated reputation scores, and how the items are ranked. Hence, we conducted two experiments in this research. The first experiment is to predict an item rating using the item reputation score generated by reputation models. The hypothesis is that the more accurate the reputation model the closer the scores it generates to actual users' ratings. For one item, we will use the same reputation score to predict the item's rating for different users. The second experiment is to prove that the proposed method produces different results than the Naïve method in terms of the final ranked list of items based on the item reputations. If the order of the items in the two ranked lists generated by the Naïve and NDR methods is not the same, we say that our method is significant.

4.1 Datasets

The dataset used in this experiment is the MovieLens dataset obtained from www.grouplens.org, which is publicly available and widely used in the area of recommender systems. The dataset contains about one million anonymous ratings of approximately 3,706 movies. In this dataset each user has evaluated at least 20 movies, and each movie is evaluated by at least 1 user. In our experiment we split the dataset into training and testing datasets with 80% of the users used to build the training dataset and the rest are used for testing.

Three new datasets were extracted from the original dataset in order to test the different reputation models for different levels of sparsity. The sparsest dataset created has only 4 ratings per movie randomly selected from users' ratings to this movie. For the second and the third datasets, each movie has 6 and 8 randomly selected ratings, respectively. Table 2 summarize the statistics of the used datasets.

Table 2: Used datasets statistics.

Dataset	**Users**	**Ratings**
Only 4 ratings per movie (**4RPM**)	1361	14261
Only 6 ratings per movie (**6RPM**)	1760	21054
Only 8 ratings per movie (**8RPM**)	2098	27723
Complete Data Set All ratings (**ARPM**)	6040	999868

4.2 Evaluation Metrics

In the experiments conducted in this research, we select two well-known metrics to evaluate the proposed methods.

Mean Absolute Error (MAE). The mean absolute error (MAE) is a statistical accuracy metric used to measure the accuracy of rating prediction. This metric measures the accuracy by comparing the reputation scores with the actual movie ratings. Equation 7 shows how to calculate the MAE.

$$MAE = \frac{\sum_{i=1}^{n} |p_i - r_i|}{n} \tag{7}$$

p_i is the predicted value (i.e., a reputation score) for a movie i, r_i is the actual rating given by a user for the movie i, and n is the number of ratings in the testing dataset. The lower the MAE, the more accurately the reputation model generates scores.

Kendall Tau Coefficient. Kendall tau coefficient is a statistic used to measure the association between two ranked lists. In other words, it evaluates the similarity of the orderings of the two lists. Equation 8 shows how to calculate Kendall Tau coefficient, where it divides the difference between concordant and discordant pairs in the two lists by the total number of pairs $\frac{n(n-1)}{2}$. The coefficient must be in the range of $-1 \leq \tau \leq 1$, where the value of $\tau = -1$ indicates complete disagreement between two lists, and the value of $\tau = 1$ indicates complete agreement. In addition, the value of $\tau = 0$ identify that the two lists are independent.

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n\left(n-1\right)} \tag{8}$$

$$n_d = |\{(i,j)|A(i) < A(j), NDR(i) > NDR(j)\}|$$
$$n_c = |\{(i,j)|A(i) < A(j), NDR(i) < NDR(j)\}|$$

n_d is the number of discordant pairs between the two lists, while n_c is the number of concordant ones, $NDR(i)$ is the reputation score for the movie i generated using the NDR method, while $A(i)$ is the reputation score generated using the Naïve method, n is the number of items and i and j are items. The aim of using the Kendall tau coefficient method is to compute the ordering difference between the two ranked item lists generated based on the reputations computed using two different reputation models. The higher the value of τ, the more similar the two ranked lists.

4.3 Ratings Prediction

In this experiment we use the training dataset to calculate a reputation score for every movie. Secondly we will use these reputation scores as rating prediction values for all the movies in the testing dataset and will compare these reputation values with users' actual ratings in the testing dataset. The theory is that a reputation value to an item that is closer to the users' actual ratings to the item is considered more accurate. The Baseline methods we will compare with include the Naïve method, Dirichlet reputation system proposed by Jøsang and Haller [8], and the Leniency-aware Quality (LQ) model proposed by Lauw et al. [2].

The experiment is done as a five-fold cross validation, where every time a different 20% of the dataset is used for testing. This method ensures that each user's data has been used five times; four times in training and one time in testing. We record the MAE in each round for all the implemented methods, and at the end we calculate the average of the five MAE values recorded for each reputation model. We have tested the ratings prediction accuracy using the four previously described datasets and the results are shown in Table 3.

Table 3: MAE results for the 5 fold rating prediction experiment

Dataset	**Naïve**	**LQ**	**Dirichlet**	**NDR**	**NDRU**
4RPM	0.5560	0.5576	**0.5286**	0.5614	0.5326
6RPM	0.5610	0.5628	0.5514	0.5608	**0.5498**
8RPM	0.5726	0.5736	0.5705	0.5693	**0.5676**
ARPM	0.7924	0.7928	0.7928	**0.7851**	0.7853

The four datasets we use include three sparse datasets (i.e., 4RPM, 6RPM, and 8RPM) and one dense dataset (i.e., ARPM). The three sparse datasets reflect different levels of sparsity. In Table 3, the MAE results using the sparsest dataset 4RPM shows that the best prediction accuracy was produced by the Dirichlet method. The reason is because the Dirichlet method is the best method among the tested 5 methods to deal with the uncertainty problem which is especially severe for sparse datasets. The proposed enhanced method NDRU achieved the second best result which is close enough to the Dirichlet method result with a small difference, indicating that NDRU is also good at dealing with uncertainty. However, when we use less sparse datasets 6RPM and 8RPM, the proposed NDRU method achieved the best results.

The last row in Table 3 shows the results of ratings prediction accuracy using the whole MovieLens dataset (ARPM) which is considered a dense dataset. We can see that the proposed method NDR has the best accuracy. Moreover, our enhanced method NDRU achieved the second best result with an extremely small difference of 0.0002. In contrast, the other baseline methods do not provide any enhancement in accuracy over the Naïve method on the dense dataset.

From the results above, we can see that the NDR method produces the best results when we use it with dense datasets, and that the Dirichlet method is the best with sparse datasets. Most importantly, the NDRU method, provides good results in any case, and can be used as a general reputation model regardless of the sparsity in datasets.

4.4 Comparisons of Item's Ranking

In this experiment, we will compare two lists of items ranked based on their reputation scores generated using the NDR method and the Naïve method. The purpose of this comparison is to show that our method provides relatively different ranking for items from the Naïve method.

The experiment is conducted in 20 rounds, with different percentage of data used every time. In the first round we used a sub-list with only the top 1% of the ranked items in one list to compare with the 1% items in the other list. The number of comparisons is equal to $\frac{n(n-1)}{2}$, n is the number of items in the top 1% of each list. For The other 19 rounds we used the top $5\%, 10\%, 15\%, \ldots, 100\%$, respectively. The reason for choosing different percentages of top items is to see the difference between different percentages of top items. Usually the top items are more influential or crucial to users.

From Fig. 2 we can find that, for all datasets, the more the items taken from the lists, the more similar the order of the items in the lists generated by the two methods. However, usually users are more interested in the top items. Therefore, the order of the top items in the lists is more crucial. If we only look at the top 20% items, we can find that the behaviour of using the whole dataset ARPM (which is much denser than the other three datasets) is different from using other three sparse datasets. For the dense dataset, the similarity reaches its minimal when we only compare the top 1% items and the similarity increases when we compare larger portions of the dataset. This result indicates that for the dense dataset, the proposed method NDR ranks the top items in the item list differently from the item list generated by the Naïve method. On the other hand, with the sparse datasets, the ranking on the top 1% of the items shows high similarity between the two lists, which indicates that the top 1

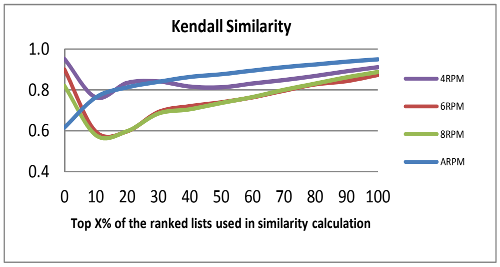


Fig. 2: Kendall similarities between (NDR) method and Naïve method using four different datasets.

5 Conclusions and Future Work

In this work we have proposed a new aggregation method for generating reputation scores for items or products based on customers' ratings, where the weights are generated using a normal distribution. The method is also enhanced with adding uncertainty part by adopting the idea of the work proposed by Jøsang and Haller [8]. The results of our experiments show that our proposed method outperforms the state-of-the-art methods in ratings prediction over a well-known dataset. Besides, it provides relatively different ranking for items in the ranked list based on the reputation scores. Moreover, our enhanced method proved to generate accurate results with sparse and dense datasets. In future, we plan to use this method in different applications such as recommender systems. Besides, this method can be combined with other weighted average reputation models that use time or user reputation in order to improve the accuracy of their results.

References

1. Resnick, P., Kuwabara, K., Zeckhauser, R. Friedman, E.: Reputation Systems. Communi-cations of the ACM, 43(12), pp. 45-48. (2000).
2. Lauw, H. W., Lim, E. P., Wang, K.: Quality and Leniency in Online Collaborative Rating Systems. ACM Transactions on the Web (TWEB), 6(1), 4. (2012).
3. Grimmett, G., Stirzaker, D.: Probability and random processes. Oxford university press.(2001).
4. Garcin, F., Faltings, B., Jurca, R.: Aggregating Reputation Feedback. Proceedings of the First International Conference on Reputation: Theory and Technology, pp. 62-74. (2009).
5. Jøsang, A., Ismail, R., Boyd, C.: A Survey Of Trust And Reputation Systems For Online Service Provision. Decision Support Systems, 43(2), pp. 618-644. Elsevier. (2007).
6. Shapiro, C.: Consumer Information, Product Quality, And Seller Reputation. The Bell Journal of Economics,13(1), pp. 20-35. RAND. (1982).
7. Ayday, E., Lee, H., Fekri, F.: An Iterative Algorithm For Trust And Reputation Management. Proceedings of the International Symposium on Information Theory, pp. 2051-2055. IEEE. (2009).
8. Jøsang, A. and Haller, J.: Dirichlet Reputation Systems. Proceedings of the Second International Conference on Availability, Reliability and Security, pp. 112-119. IEEE. (2007).
9. Leberknight, C. S., Sen, S., Chiang, M.: On The Volatility Of Online Ratings: An Empirical Study. E-Life: Web-Enabled Convergence of Commerce, Work, and Social Life, pp. 77-86. Springer Berlin Heidelberg. (2012).
10. Riggs, T., Wilensky, R.: An Algorithm For Automated Rating Of Reviewers. Proceedings of the First ACM/IEEE-CS joint conference on Digital libraries, pp. 381-387. ACM.. (2001).
11. Jøsang, A., Ismail, R.: The Beta Reputation System. Proceedings of the 15th bled electronic commerce conference, pp. 41-55. (2002).
12. Sabater, J., Sierra, C.: Reputation And Social Network Analysis In Multi-Agent Systems. in Proceedings of the first international joint conference on Autonomous agents and multiagent systems. pp. 475-482. (2002).
13. Bharadwaj, K. K., Al-Shamri, M. Y. H.: Fuzzy Computational Models For Trust And Reputation Systems. Electronic Commerce Research and Applications, 8(1), pp. 37-47, (2009).