A study on rating services based on users' categories

Gianpiero Costantino, Fabio Martinelli, Marinella Petrocchi
IIT–CNR, Via G. Moruzzi 1, 56124 Pisa, Italy
Email: name.surname@iit.cnr.it

***Abstract*—In the Internet age, people are becoming more and more familiar in experiencing online services. In many cases, the customer commits herself and her assets in a business transaction with no (or limited) possibility to test the service/good she is booking/buying. Hence, there is the need to prove the trustworthiness of such services for supporting a user in her choice. Many websites feed the customer with reviews of past users representing their degree of satisfaction. In this paper, we consider a scenario where different services may be grouped together to form packets, and we design and implement a simple procedure through which a customer can choose the packet that best satisfies her expectations. The final choice will be driven both by the qualities of the reviews on the constituting services, and by the customer's personal preference and attitudes. To automatise the procedure, we survey real behaviours of users when they choose a service and give reviews, by obtaining a probabilistic model plugged in our simulator. In particular, we deal with the issue of false review, reported by unfair users that intentionally act malevolently. The simulations results show that our system is robust enough up to a certain number of unfair feedback.**

***Keywords*-Reviewing Systems, Design and Evaluation, Probabilistic Client Model, Unfair Feedback.**

I. Introduction

The availability of a large pool of e-services may lead consumers to face the difficulty of choosing the one(s) that satisfy at best their needs. What generally helps in such situations is a service provider in charge of delivering a list of services, decorated with additional criteria supporting the consumer in her choice. A natural support is represented by rating services, *e.g.* by attaching numerical scores, or textual judgments, summing up the degree of satisfaction of past users towards that service. High scores will encourage the consumer in making her choice, even if the final selection will be influenced also by personal preferences (*e.g.* users will not always choose the hotel with the highest score, as it is probably one of the most expensive).

Here, we consider a scenario in which a broker provides a set of services to different kind of clients. We propose a procedure for rating services through review computation and a simple protocol to offer the composition that best satisfy the client's needs. For our prototype, we rely on a probabilistic client model obtained by reproducing the behaviour of real clients when they give feedback and when they choose services. For designing and implementing such a model, we gather and analyse data from two popular websites. We validate the model through simulations, aimed at testing how the system works in presence of unfair clients that intentionally provide false reviews, a frequent misbehaviour confirmed by recent studies, see, *e.g.* [12].

The paper is organized as follows. Section II recalls related work in the area of rating systems. In Section III, we describe the reference scenario, the procedure for review computation, and the protocol for requesting and experiencing packets of services. Subsection IV-A shows how we derive a probabilistic model both for the client choice and the client feedback. In Subsection IV-B, we present a number of evaluations we have carried out. Finally, Section V concludes the paper.

II. Related Work

The rating of a service (or a product) is kept up-to-date according to algorithms generally built on the principle that the new rating is a function of the old ratings and the most recent review(s) [9]. In simple models, such the one adopted by Ebay prior to May 2008, past and new ratings about the outcome of online transactions between a buyer and a seller contribute in an equal manner to the calculation of the trustworthiness of the seller. More recently, Ebay started considering only the percentage of positive ratings of the last twelve months. The same temporal window is also used in the Amazon marketplace. Other models combine in a weighted mean the old rating and the newest reviews. Proposals to evaluate such weights are based on, *e.g.* the trustworthiness of the reviewer [1], [3], [17], the evaluation of the users satisfaction for a set of parameters characterising the object [8], the review freshness, or the distance between the single review and the overall score (as suggested in [9]). Other work, like in [2], [15], suggests to weigh according to the users' expertise, in order for instance to weigh more the reviews given by professionals and less the reviews given by regular users. In our approach, ratings are assigned according to categories of users, as commonly classified in popular websites specialised in services advice. The proposed reviewing system is parametric with respect to the weights to be assigned to past and new feedback. In particular, in this paper, we propose a configuration that is optimal, at least for our scenario, with respect to a percentage of unfair ratings and the speed in achieving reviews values comparable with a set of reference values.

Online reviews posted by users should be considered truthful if supported by a reputation mechanism assessing

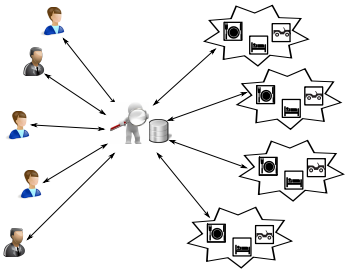


Figure 1. Reference Scenario: Clients-Broker-Packets

the trustworthiness of the reviewers. We acknowledge research work in the area of immunising reviewing systems against unfair (or incomplete) ratings, *e.g.* [4]–[7], [10], [16], [18]. In particular, work in [11] introduces a new definition of unfairness, by considering two categories of advisers, the first category representing users that intentionally act malevolently, while the second one representing users that lack of sufficient experience for correctly giving advice. This differentiation allows the authors to propose a two layered filtering algorithm that first detect newcomers with lack of experience, and then classify the remaining advisers according to their credibility. In this paper, rather than proposing a way to cut off unfair ratings, we investigate how robust our reviewing system is, in presence of a certain percentage of unfair ratings.

Work in [19] focuses on feedback selection, and proposes an algorithm to filter past feedback that matches best a user's context. The framework has been tested with real consumers to test its accuracy. Here, instead of dealing with real consumers, we analyse review sets collected from real websites, in order to automatise the behaviours of real users.

Finally, it is worth noticing that recommendation systems have been successfully adopted within large-scale agent-based (social) networks for the selection of trading partners and useful items: as an example, the author of [14] proposes a tag-based recommendation system that maximises some utility function of the users. Also, work in [13] considers how to enable social evaluations and proposes the integration of a cognitive agent and a cognitive reputation model allowing the agent to take decisions in a multi-context environment based on beliefs, desires, intentions, and plans. We acknowledge this area of research as a relevant mean to trade off the subjective attitude of the user's opinion and the community's opinion.

III. Architecture and Scenario

Fig. 1 illustrates our reference scenario, in which an online service broker B provides a list of *packets* P to a client C. Each packet consists of constituting services S^i. As a simple running example, we assume that the client is a traveller willing to book a trip via B. So, C requests accommodation, transportation, and refreshment, and each packet P^j will consist of: hotel S^j_H, car rental S^j_{CR}, and restaurant S^j_R[1]. Hereafter, we let a review range over the set $\{1, \ldots, 5\}$ of real numbers.

The procedure for requesting and experiencing a packet is quite simple:

1) C asks the broker a packet (hotel + restaurant + car).
2) B presents a list of packets, sorted according to the client's preferences. The way in which such preferences are evaluated is explained in Section IV-A1.
3) C chooses the packet whose review best matches her preferences (see Section IV-A1), experiences P, and gives feedback on the services constituting P.
4) B updates the reviews of the single services, and forms a new list of packets for the next client.

We focus on step 4, *i.e.* the computation and updating of the services' reviews. Not surprisingly, we think that the new value should depend both on the more recent reviews and on reviews due to experiences of past users. The following formula generically indicates that the new review is a function f of the old reviews and the last one.

$$R^S_{new} = f(R^S_{last}, R^S_{old})$$

In particular, we propose the next quite simple formula, where R^S_{last} denotes the last review on the service S, R^S_{old} is the old review, and w_{last}, w_{old} are weights ranging over $\{0, \ldots, 1\}$ and $w_{fb} + w_{old} = 1$.

$$R^S_{new} = R^S_{last} * w_{last} + R^S_{old} * w_{old} \quad (1)$$

The weights are opportunely tuned in order to give more or less importance to history rather than to new feedback.

IV. Validation

In this section, we first characterise in a specific way each actor involved in our scenario. Then, we propose a way to characterise the clients' preferences, in order for the broker to propose to each client the list of packets in which the first one is the closest to that client's preferences. Also, we present how we derive values R^S of clients' reviews of expression 1. Finally, we propose a number of experimental results, for validating such formula in presence of a percentage of unfair clients that report false reviews.

Our scenario involve a set of clients, a broker, and a set of e-services. In particular:

- The broker is an agent that interfaces services and clients, by following the protocol given in section III.
- The services are hotels, restaurants, and car rentals.
  - In order to validate our proposal, we need reference values for the review of each service in a steady state. For each service, we take as the set of

[1]Here, we simplify the scenario, by considering that all the possible accommodation services (resp., transportation/refreshment services) are represented by a hotel (resp., a car rental/a restaurant).

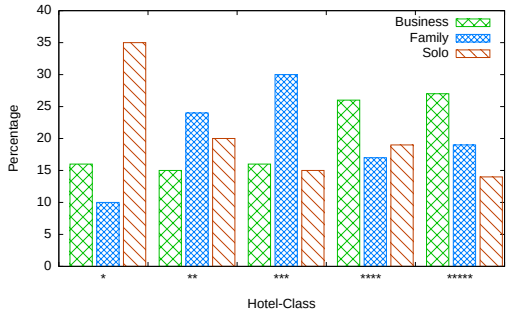


Figure 2. NYC hotels: preference of clients. Percentage of clients choosing NYC hotels, per client typology and hotel class

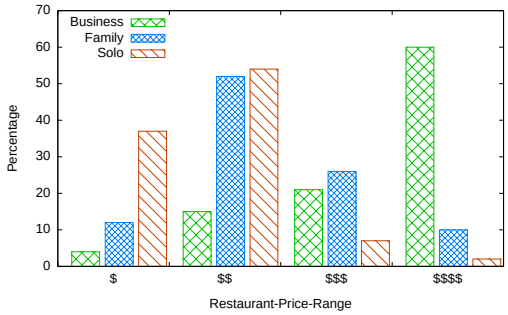


Figure 3. NYC restaurants: preference of clients. Percentage of clients choosing NYC restaurants, per client typology and restaurant price range

review reference values that one surrounding the category reported in the website. As an example, a reference review value for a 5 star hotel ranges over $\{4.51 \ldots, 5\}$, and for a 4 star hotel over $\{3.51 \ldots, 4.5\}$.

- Each of the services enters the system with an initial random review value. We justify this choice to test the goodness of our proposal, in terms of proving: 1) if the review values come to results comparable to the reference values (see above); 2) how fast the review computation mechanism is in adjusting the initial random values.

- We consider three categories of clients: solo traveller C_{st}, family C_f, and businessman C_b.

A. Client Model

1) Modeling a client preference: As introduced in Section III, the broker proposes a ranking of packets sorted according to the client's preferences. We assume that three *preference values* are dynamically associated to each client, namely v_h for hotel, v_{rc} for car rentals, and v_r for restaurants. All these three values range over $\{1, \ldots, 5\}$.

We propose to calculate the preference values v_i by considering behaviours of real clients. In particular, we examine popular websites offering travel advices about hotels, restaurants, and car rentals[2].

Regarding hotels, we consider a subset of the 430 hotels in New York City reviewed on Tripadvisor.com. This website allows a user to filter clients' categories, in order to visualize, *e.g.* how many past users of a given category has chosen a particular hotel. Fig. 2 shows the results obtained by our survey. For example, we obtain that, on average, about 27% of the Tripadvisor businessmen users prefer a 5 star hotel, 26% of them choose a 4 star hotel, 16% stay at a 3 star hotel, while 15% and 16% choose, respectively, a 2 star and 1 star hotel.

Regarding restaurants, we consider a subset of the almost 7000 restaurants in New York City revised on Tripadvisor. The website distinguishes them according to the price range, between $ and $$$$. We survey how many businessmen, solo travellers, and families have chosen a restaurant that falls within a particular price range, over a period of time. This leads to the results shows in Figure 3, where it is possible to see that, for example, 60% of businessmen considered in our survey prefer a $$$$ restaurant.

Finally, we consider the website viewpoints.com, giving advice on best car rentals (www.viewpoints.com/Rental-Cars). We notice that the majority of car rentals have a similar number of reviews, meaning that they have been chosen with a similar frequency.

We suggest to assign to each client a preference value v_i in a probabilistic way. For example, a solo traveller will have attached v_h = 1 with probability 35%, = 2 with probability 20%, = 3 with probability 15%, and so on (see figure 2). The same reasoning holds for preference values v_r for restaurants, while, given the results of our survey, we decide to attach to each client v_{rc} = 1 with probability 20%, = 2 with probability 20%, *etc.*. .

Now, we can clarify the way in which the broker sorts the list of packets according to the clients' preferences. Suppose that a businessman asks for a packet (step 1 in the procedure of Section III). The broker will first assign to that businessman v_h^{bus}, v_r^{bus}, and v_{cr}^{bus} in a probabilistic way, according to the results of a survey similar to ours. Then, B will consider the hotel, the restaurant, and the car rental that have obtained reviews closest to v_h^{bus}, v_r^{bus}, and v_{cr}^{bus}, and they will form the packet ranked first. Subsequently, the broker selects the hotel, the restaurant, and the car rental with the second closest values of reviews, and these will form the second packet, *etc.*. . The numerical closeness is in absolute value. Figure 4 shows an example of a list prepared for a client of category *businessman* whose preference values are v_h^{bus} = 4, v_r^{bus} = 4, and v_{cr}^{bus} = 3. As we can see, the first packet is the one whose components have obtained the review values closest to the client's preference values.

2) Modeling a client review: Once the client has experimented the packet, the broker asks her to provide some feedback. In order to automatise the review computation,

[2]All the surveys refer to data gathered from websites in fall 2011.

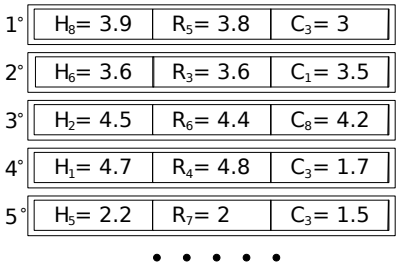


Figure 4. A list of packets with reviews sorted following the client's preferences. The example shows the list for a businessman with v_h =4, v_r =4, and v_{cr} =3.

we propose a probabilistic feedback model, based on real advices published on Tripadvisor.com. We consider restaurants and hotels in New York City.

On Tripadvisor, each hotel has a set of associated reviews. Reviewers can judge a hotel with five marks: Excellent, Very good, Average, Poor, Terrible. Reviews may be filtered per client typology, *e.g.* businessmen, families, and solo travellers. Fig. 5 shows the distribution of feedback, per client typology and hotel class. As an example, considering the NYC 5 star hotels, on the totality of 613 businessmen reporting reviews, 393 give an Excellent mark (64%), 92 businessmen a Very good mark (92%), 72 an Average mark (12%), 35 a Poor mark (6%), and 21 a Terrible mark (3%).

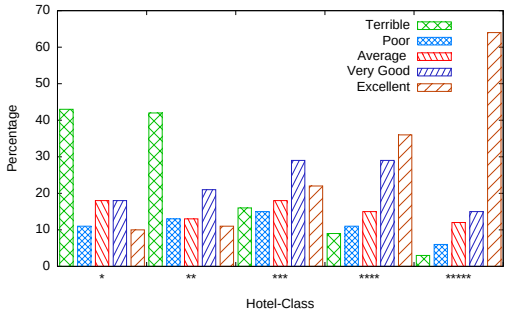
Tripadvisor does not allow to filter restaurant reviews according to the client's typology. Thus, we consider a generic traveller. Results of our survey are illustrated in Fig. 6. As an example, we can see that 44% of clients consider a 4$ NY restaurant Excellent, 33% give a Very good mark, 17% think that 4$ NY restaurants are on Average, and 4% and 2% are unsatisfied, giving Poor and Terrible marks.

Finally, reviews on car rentals were not sufficient to derive a feedback distribution. Thus, we decide to consider a uniform distribution of feedback, ranged over $\{1.0,\ldots, 5.0\}$.

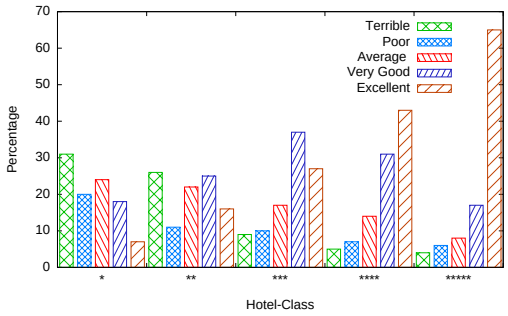
In our system, each service is associated to a default classification (*e.g.* restaurants are classified by price range, and hotels are classified by stars). When a restaurant (respectively, a hotel) is evaluated, a client feedback is probabilistically obtained according to the percentages given in Fig. 6 (respectively, Fig. 5).

For example, a 4$ restaurant is judged *Excellent* with a probability of 44%, *Very good* with a probability of 33%, *Average* 17% and so on. Since we consider as review values real numbers ranged over $\{1, \ldots, 5\}$, such textual feedback are uniformly mapped to numerical values in intervals as in Table I. These values are the R^S values of expression 1.

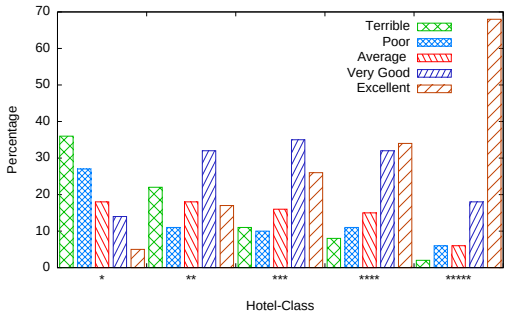
3) Unfair clients: Typically, reviewing systems can be altered intentionally by unfair clients. Goal of these users is to post false reviews in order to penalise services. A trivial model is represented by clients who give feedback in a completely random way.



(a) NY hotels: Business feedback



(b) NY hotels: Families feedback



(c) NY hotels: Solo travellers feedback

Figure 5. NYC hotels: Clients' feedback. Percentage of clients giving a certain feedback, per client typology and hotel class.

Table I

Mark	Feedback Values
Excellent	[4.51 , . . ., 5.0]
Very good	[3.51 , . . ., 4.5]
Average	[2.51 , . . ., 3.5]
Poor	[1.51 , . . ., 2.5]
Terrible	[1.0 , . . ., 1.5]

We tackle this issue by considering unfair clients and observing how our system reacts. Here, we adopt a model for the attacker that gives reviews in a probabilistic fashion, and we consider the distribution function got from our Tripadvisor survey, but in a mirror-like fashion. According to the trend shown in the figures, an *Excellent* mark is given to

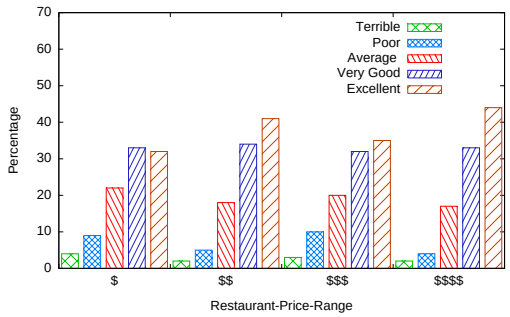


Figure 6. NYC restaurants: Clients' feedback. Percentage of clients giving a certain feedback, per restaurant price range.

a high-level service (*e.g.* a 5 star hotel) with high probability. Following the mirror-view strategy, an unfair client gives a *Poor* mark with that same probability.

B. Experimental Results

We present some experimental results obtained through a study aiming at characterising the behaviour of our reviewing system. The study is performed implementing an ad-hoc simulator that mimics our framework by letting: 1) the broker propose the list of packets to each client, according to their category and preference values (see section IV-A); 2) the client choose and experience a packet; 3) the feedback be given to each service according to the client's feedback model (see section IV-A2); 4) the broker update the reviews of constituting services according to new and old feedback, following expression 1 of Section III. A number of different interactions is realised in subsequent steps.

The simulator has been developed in JAVA (www.java.com), it can be easily run on traditional laptops or desktop computers and is available online[3]. We ran several simulations with different values for w_{old} and w_{last} (see expression 1 in Section III). Tuning the weights, more relevance is given to past feedback R_{old} or to new feedback R_{last}.

1) Fair Clients: Figures 7(a)-7(b) show the review trend in a setting where all clients provide fair feedback. We simulate 2000 interactions: in each of them a client chooses a packet according to her preferences, she experiences and she gives feedback according to her feedback model. Starting by initial random reviews, the services quite quickly obtain reviews very close to the *reference values*. For example, reference values for high class hotels and restaurants are in $\{4.5, \ldots, 5\}$. We can see that the reviews quickly come to comparable values.

2) Unfair Clients: We aim at finding the optimal weights in expression 1 in order to suffer as less as possible from unfair feedback. Thus, we ran several simulations, with different values for weights and different percentages of unfair clients. We consider those in Table II.

[3]http://www.iit.cnr.it/staff/gianpiero.costantino/CNR-PersonalPage/Simulator.html

Table II
PERCENTAGES OF FAIR/UNFAIR CLIENTS

Fair	Unfair
100	0
90	10
80	20
70	30
60	40
50	50

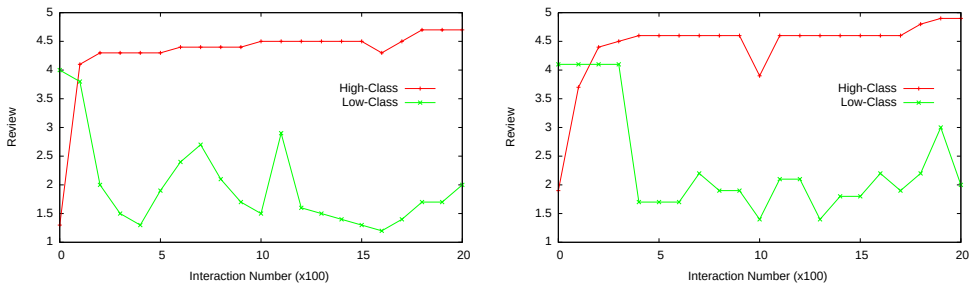
In Figure 8 we show the most relevant results we have obtained, for a 4 star hotel. On the left column, the review trend is shown in a setting with a low amount of unfair clients (up to the 20% of the totality), while in the right column a higher percentage is considered (up to 50%).

Giving more importance to new feedback, the trend is less stable. Indeed, few new positive (resp., negative) feedback are sufficient for rapidly increasing (resp., decreasing) the service's review values. Hence, an attacker may easily compromise a service, see, *e.g.* Fig. 8(a), and above all, Fig. 8(b), where it is possible to see that a relevant amount of unfair clients can provoke a completely distorted review value. On the other hand, when using very low weights for new feedback (*e.g.* $w_{last} = 0.1, w_{old} = 0.9$, Figures 8(c)-8(d)), the resulting trend is flatter. A flatter trend may affect the disclosure of suspicious behaviours.

The best trade off that we have found between w_{last} and w_{old} is presented in Figures 8(e) and 8(f). A higher importance is given to old feedback. Nevertheless, new interactions are properly considered ($w_{last} = 0.3$ and $w_{old} = 0.7$). Figure 8(f) highlights that these values of w_{last} and w_{old} allow our system to be quite robust even in presence of a high percentage of unfair clients. Indeed, the resulting trend is not affected by substantial modifications.

V. CONCLUSIONS

We have proposed a rating system for online services. In order to automatise the procedure of review computation, we first collected data from popular websites specialised in clients' reviews. From the analysis of such data, we then derived a probabilistic model of feedback for three kinds of clients: businessmen, families, and solo travellers. The efficacy of the model has been evaluated by simulating a system able to get, as input, feedback of past clients, distributed according to the model that we have derived, and return the updated review value. Simulations show that our mechanism works well up to a certain number of unfair feedback. Also, in our scenario, different kind of services can be composed together and they form packets. Packets are offered to clients according to her preferences, here derived from the analysis of real behaviours of users when they make choice on the Internet.



(a) Hotel review ($w_{old} = 0.4$, $w_{last} = 0.6$)

(b) Restaurant review ($w_{old} = 0.4$, $w_{last} = 0.6$)

Figure 7. Review trend for two distinct services.

The surveys that we have carried out considers a relatively small number of clients, services, and clients' typologies, but this modeling way could be easily adopted in real world implementations, since many websites specialised in services' reviews usually rely on huge datasets.

We think that other interesting directions could be investigated. First, unfair feedback may lead to a complete distorted review value. Our work could be extended with a proactive component where alarms are raised when something is suspected to go wrong. Secondly, assuming that services initially enter the system with an initial review value fixed in accordance with a broker in a business agreement, anomalies between that value and the value calculated with the reviewing system may lead to re-considering the agreement. We leave this for future work based on contracts.

REFERENCES

[1] S. Buchegger and J. Le Boudec. A robust reputation system for mobile ad-hoc networks. Technical report, IC/2003/50 EPFL-IC-LCA, 2003.

[2] W. Chen, Q. Zeng, and L. Wenyin. A User Reputation Model for a User-Interactive Question Answering System. In *International Conference on Semantics, Knowledge and Grid*, page 40. IEEE, 2006.

[3] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. Choosing reputable servents in a p2p network. In *World Wide Web*, pages 376–386. ACM, 2002.

[4] C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *ACM Conf. on Electronic Commerce*, pages 150–157, 2000.

[5] C. Dellarocas and C. A. Wood. The sound of silence in online feedback: Estimating trading risks in the presence of reporting bias. *Management Science*, 54(3):460–476, 2008.

[6] Q. Feng, Y. Yang, Y. Sun, and Y. Dai. Modeling attack behaviors in rating systems. In *Distributed Computing Systems Workshops*, pages 241 –248, 2008.

[7] J. Gorner, J. Zhang, and R. Cohen. Improving the use of advisor networks for multi-agent trust modelling. In *Privacy, Security and Trust*, pages 71 –78, 2011.

[8] N. Griffiths. Task delegation using experience-based multi-dimensional trust. In *AAMAS*, pages 489–496. ACM, 2005.

[9] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.*, 43:618–644, 2007.

[10] S. Liu, C. Miao, Y.-L. Theng, and A. C. Kot. A clustering approach to filtering unfair testimonies for reputation systems. In *Autonomous Agents and Multiagent Systems: Vol. 1*, pages 1577–1578, 2010.

[11] Z. Noorian, S. Marsh, and M. Fleming. Multi-layer cognitive filtering by behavioral modeling. In *Autonomous Agents and Multiagent Systems - Volume 2*, pages 871–878, 2011.

[12] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Human Language Technologies*, pages 309–319, 2011.

[13] I. Pinyol, J. Sabater-Mir, P. Dellunde, and M. Paolucci. Reputation-based decisions for logic-based cognitive agents. *Autonomous Agents and Multi-Agent Systems*, 24:175–216, 2012.

[14] S. Sen. Finding useful items and links in social and agent networks. In *Agents and Data Mining Interaction*, volume LNCS 5980, page 3. Springer, 2010.

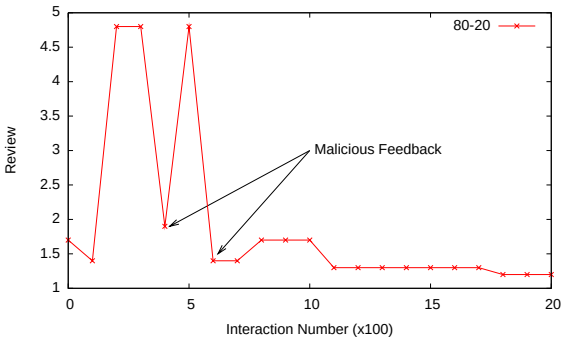
[15] T. van Deursen, P. Koster, and M. Petkovic. Hedaquin: A reputation-based health data quality indicator. *Electr. Notes Theor. Comput. Sci.*, 197(2):159–167, 2008.

[16] A. Whitby, A. Jsang, and J. Indulska. Filtering out unfair ratings in bayesian reputation systems. In *Workshop on Trust in Agent Societies*, 2004.

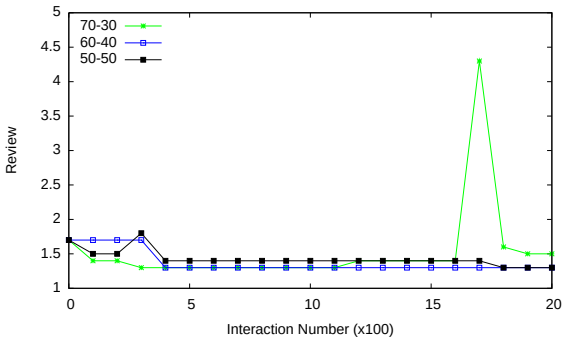
[17] B. Yu and M. P. Singh. Detecting deception in reputation management. In *AAMAS*, pages 73–80. ACM, 2003.

[18] J. Zhang and R. Cohen. Trusting advice from other buyers in e-marketplaces: the problem of unfair ratings. In *ICEC*, pages 225–234, 2006.

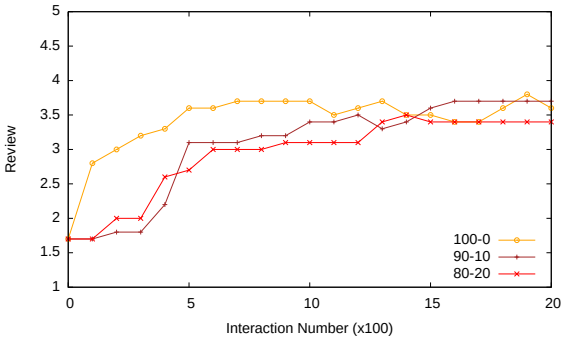
[19] W. Zhao et al. A user-oriented approach to assessing web service trustworthiness. In *ATC*, LNCS 6407, pages 195–207. Springer, 2010.



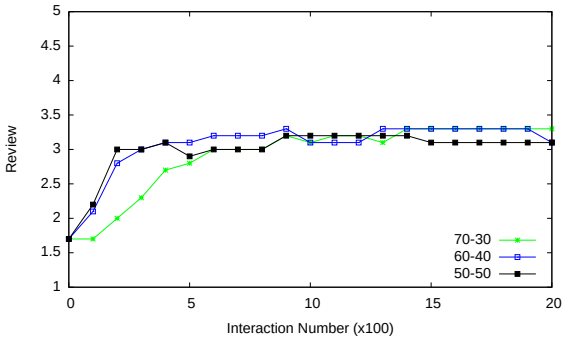
(a) Weights: $w_{last} = 0.8$ and $w_{old} = 0.2$



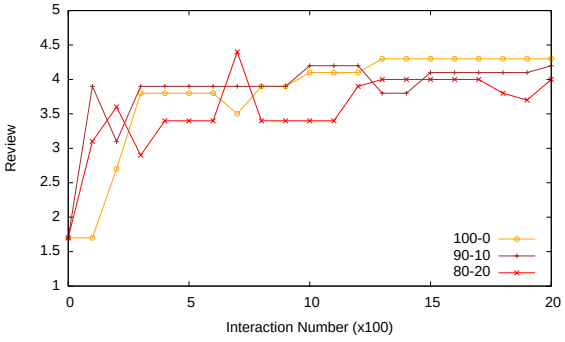
(b) Weights: $w_{last} = 0.8$ and $w_{old} = 0.2$



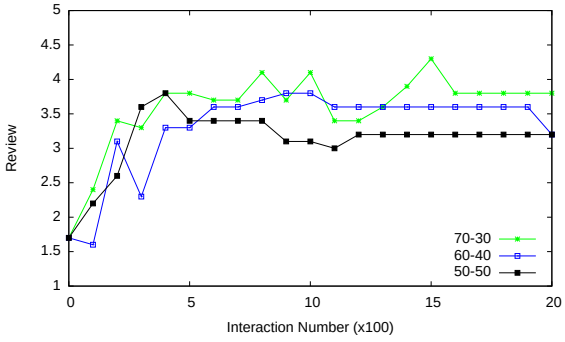
(c) Weights: $w_{last} = 0.1$ and $w_{old} = 0.9$



(d) Weights: $w_{last} = 0.1$ and $w_{old} = 0.9$



(e) Weights: $w_{last} = 0.3$ and $w_{old} = 0.7$



(f) Weights: $w_{last} = 0.3$ and $w_{old} = 0.7$

Figure 8. High-class hotel: Review trend varying w_{last}, w_{old}, and the percentage of unfair clients.