2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns

An Enhanced Framework for Web Recommenders

Oana Dini
University of Besançon, France
oana.dini@univ-fcomte.fr

Pascal Lorenz
University of Haute Alsace, France
lorenz@ieee.org

Hervé Guyennet
University of Besançon, France
guyennet@lifc.univ-fcomte.fr

Abstract – **The web-based transactions, web services, and service oriented platforms require mechanism to announce, select, and use different services. There is a dilemma of 'use and trust' or 'trust and use' for different services based on the notion of reputation. Indirect servicing makes it difficult to really assess a given service or provider. The paper presents a framework and appropriate mechanisms to evaluate the services/providers in the light of their respective direct impact on user perception. It is shown that the proposed mechanism allows capturing situations usually not possible to be captured in the current approach. The technique has a major impact on e-commerce systems, on systems based on service-oriented architecture, and on all auction-based transactions.**

Keywords – **recommenders, reputation, indirect reputation.**

1. Introduction

With the overwhelming amount of information, products, and services available over the Internet, it has become harder for the users to select the ones that fit best their needs or requirements. First of all, it is too difficult and time consuming to sort through hundreds of items and select the needed one. Also, there is the problem of trusting the provider for that item and not only that, but trusting that the provider is offering a product that meets the user's requirements. In order to assist the user in selecting the product or service that it needs, recommender systems have been proposed.

Recommender systems (RS) have been the subject of many studies and products over the last decade. The term was first brought up by Resnick and Varian [1], which, as mentioned in [2], it was mostly a replacement for "collaborative filtering" proposed in [3].

Recommender systems are defined as systems, which collect ratings from users and then analyze the data to produce recommendations to other users [4]. There are several techniques used to generate recommendations, but the main categories are Content-based Filtering (CBF), Collaborative Filtering (CF), and Hybrid approaches [5].

RS are important in electronic commerce, especially for marketing [6] and they have been widely used in order to attract and retain customers. The relation between the loyalty of users and RS was studied in [7] using data from Amazon.com. Their findings showed that the presence of consumer reviews helps with retaining customers and also attracting new ones. In time, the business gains reputation, which usually translates to increase in business.

There are a few challenges in optimally using the recommenders due to the variety of user's profile and its volatility and the reputation of different service providers. For dealing with these aspects, recommenders usually use product rating, confidence in service providers, and regularly update this information for an accurate suggestion for a given request.

In all the existing approaches, some unprovable assumptions are considered for the purpose of easily computing reputation. Aspects like partial feedback, ignorance of customer confidence, and most importantly lack of information on the service provider identity are major challenges for an accurate reputation per product, per service provider, per context, per user profile.

In this paper, we propose an approach taking into consideration the above challenges and deriving mechanisms for a more accurate reputation considering direct and indirect product delivery.

The remainder of the paper is as follows; Section 2 presents basic concepts and major achievements on recommender implementations. The proposal of an enhanced framework for an accurate reputation is presented in Section 3. A use case is presented in Section 4, while conclusion and future work are discussed in Section5.

2. Related Work

As the proposed approach touches the recommendation and reputation on recommenders, service providers, and products, we first introduce some basic concepts.

978-0-7695-3862-4/09 $26.00 © 2009 IEEE
DOI 10.1109/ComputationWorld.2009.124

2.1 Concepts

The core information of a recommender is a list of offers (products) and ratings of those products based on feedback received after a series of recommendations. The *rating* is subject to incomplete, fictitious feedback, volume of transactions for a given product or provider, and confidence in feedback. Based on the ratings, the recommender computes its own *ranking* per product.

s[r], P[r] represents a service or a provider with the rank r, where r is an integer.

Associated with the ranking is the notion of *reputation* that in fact determines the ranking. The reputation formula, while product oriented, it might not be accurate, as its computation cannot avoid some realities, such as some service providers have private relationships with recommenders (e.g., publicity, sponsorship) or indirect servicing (recommended product might not be produced by the front end provider, but simply delivered by it).

Reputation is an index associated with the service or a product based on user feedback that is taken into consideration when the ranking is calculated. The reputation index usually belongs to a **set, {outstanding, very good, good, acceptable, bad}.** A recommender might increase the rank of a service when its reputation index, for example, passes from very good to outstanding [8]

Similarity is another concept used in generating recommendations. In order for a recommender to suggest products to a user, it needs to find a commonality among users (this applies in the collaborative approach) or among the products that were rated in the past by the user (this applies in contend-based approach). There are different techniques used to compute the similarity measure, but the most used are correlation-based and cosine-based techniques [5] [9]. Similarity is an index associated with two services or products. For example, $\mathbf{s_1}$ **[~/80%]** $\mathbf{s_2}$ means s_1 is similar with s_2 with an acceptance of 80% based on the service's features or in the same range of ranking.

2.2 Current approaches for recommenders

Recommenders are usually classified based on the approach for making the recommendations. There are three main categories of recommender types: Content-based Filtering, Collaborative Filtering, and Hybrid Filtering.

The *Content-based Filtering* recommends to users items that are similar to the ones searched by the users in the past [5][9]. This type of recommendation technique is mostly used to recommend text-based items such as documents and newspapers. In order to produce the recommendations, the system needs a profile of the user, which is represented by a set of terms. The profile can be obtained from the user through a questionnaire or it can be learned from their past transactions. This type of filtering has its shortcomings. Since it is content-based, it needs to have the representation of data in a matter that can be machine-parsable (e.g., article). It is harder to apply this technique in the case of movies, music, images, which are not machine-parsable.

The *Collaborative Filtering* (CF) [3] tries to predict the relevance of an item based on the ratings done by other users. It accumulates ratings of products and whenever a request comes, the system identifies similar users and recommends the products rated by them. In this type of filtering, the user profile is defined by a vector of items and their ratings, which is updated over time. As opposed to the CBF, this type of filtering can be applied to any kinds of items, not only to machine-parsable items. However, there are limitations with this approach, mostly caused by the lack of data points in initial stages: new user and new item.

The *hybrid algorithms* usually combine the content-based and the collaborative algorithms to overcome some of the limitations of the other two approaches. This approach has been adopted by some RS [10], [11]. There are different ways to combine the two algorithms and [5] present the different approaches in detail.

Like we mentioned above, the reputation of a business is gained in time, mainly based on reviews from users. This brings up another point and that is obtaining accurate reviews from users. Many users are not willing to leave feedback after a transaction is completed. One reason for not leaving feedback is the lack of incentives. If there isn't some kind of payoff for the feedback, the user won't put the effort into posting one. An incentive mechanism is addressed in [12] where incentives are given to users who provide honest feedback through a side payment mechanism. Examples of incentives mechanisms are Amazon's "Top Reviewers" practice and Epinions.com referral fees practice [13]. Another reason for not leaving feedback is to purposely withhold information about a product that gives its user an advantage [14].

Another concern related to the validity of the reviews is the manipulation of the reviews by parties with direct vested interest. Businesses can review their own products in order to boost the sales. Also, the competition can leave or fabricate negative feedbacks to undermine the competitor's reputation. There are ways to filter out biased feedbacks and to prevent manipulation [15], but preventing coordinated collusion attacks is still an issue. eBay for example, doesn't have a problem with feedback manipulation. The feedbacks can only be left by users who are registered with them and who made a purchase on eBay. However, if a group of users agree with a seller to leave positive feedback for fictitious auctions (e.g., the seller can post multiple 1 cent auctions on which the users can bid), the seller's ratings can be positively affected. These users are usually called shills. This approach would require quite an effort (the larger the number of shills, the bigger the impact), but it can be achieved.

Reputation is very useful in RS and eBay is one example of a reputation system that proves that their approach works well. However, having a centralized reputation system such as eBay can bring other issues, such as vulnerability and inflexibility of the system [14].

In [14], the authors propose a distributed trust and reputation management framework. The users choose a trust broker and after each transaction with a service, the user sends its rating to its trust broker. This way, the trust broker builds a reputation about a service based on the user's feedback. The brokers exchange reputation information among themselves in order to collect more information about the available services. This framework relies on the user's feedback only, ignoring the business model of the provider.

In reality, a provider may subcontract the service from somewhere else and in the end take all the credit. The question now is how to make the Recommender aware of the underlying transactions among the providers so all providers receive fair rating. If Provider 1 contracts a service from Provider 2, Provider 2 should receive credit for its service also.

3. An enhanced recommender model

In this section, we present a Recommender Model that can handle the sub-contract mechanism, yet keeping an accurate information on a given provider reputation (leading to an accurate ranking).

3.1 Setting the case

A simple scenario is presented in Figure 1, where the user is interested in service s_1 from P_1. The user asks the Recommender for the best provider for service s1 within specific parameters. The Recommender replies with either a provider that has the best reputation for service s_1 or with a list of providers $\{P_i\}$ for s_1. Let us assume P_1 is registered of being capable to deliver s_1 (others might be registered for s_1 as well). The Recommender cannot know if P_1 has the service or if it contracts it from a different provider. If P_1 is contracting s_1 from P_2, the transaction between P_1 and P_2 is transparent to both the Recommender and the user. At the end of the transaction, the user sends the rating of P_1 to the Recommender and P_1 receives all the credit for the transaction. This leads to an inaccurate reputation and altered ranking.

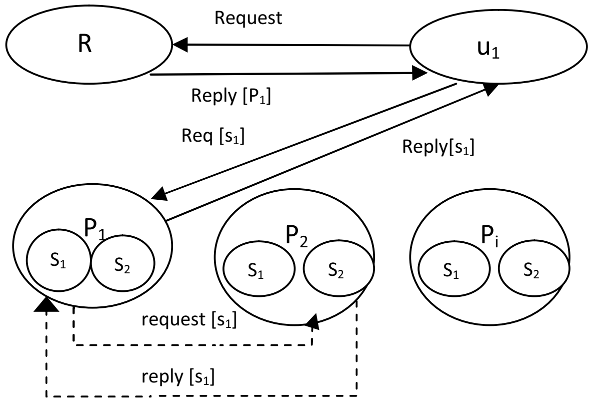

Figure 1. Indirect reputation

If the reputation of the provider is based only on the user's feedback, there is no way to assess the ultimate role of each provider. In order to have a more accurate picture of the providers' involvement, we propose that feedback from the providers be taken into account when establishing reputation. This includes both the front end provider (in our case P_1), as well as any subcontracted providers (in our case P_2). All feedback goes directly to the Recommender.

The ideal scenario would be when all the users and providers report 100% of the transactions. In reality, users don't always leave feedback and providers don't always report rendered services. In such a case, the Recommender is left to deal with an incomplete set of data. Moreover, some of the reported data may be fabricated by both users and providers.

3.2 Recommender representation model

Apart from the mechanism of collecting the feedback and interfacing with the users, the core information present in a recommender is stored in a service database. This allows a request to be replied to with a service or a list of services, eventually with a degree of similarity associated with each service. Usually, the recommender keeps information on relative ranking among these entities.

We propose an enhanced model, which takes into account the user's profile and behavior, and a list of potential providers for a given service. This allows a more refined ranking scheme where providers can be rated per service.

While ranking is based on user feedback, there is no appropriate mechanism to consider the user's expectation (e) and credibility (c). By user expectation we mean the probability of having the user leave feedback after a service was delivered. The credibility refers to the user's ability to give a trusted rating. Usually both, expectation and credibility are expressed as percentage.

In Figure 2, we present the enhanced recommender model. The recommender stores information about the available services, the providers and their services, plus the user profile, which includes its expectancy and credibility. Both services and providers are associated with a rating. The providers' rating is done within the context of a service. This way, the rating can be done per product and per provider for a specific product.

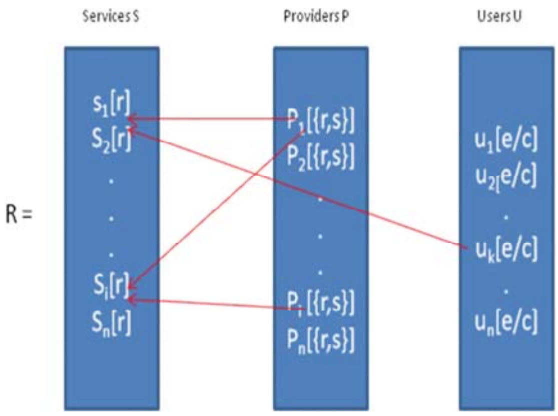

Figure 2. Enhanced Recommender Model

By keeping the relationships between the providers, their services, and also the users who requested the available services, the recommender can provide better suggestions and answer to more complex queries.

We classify queries in two categories, i.e., U-R and P-R. Some salient queries U-R might be:

Query 1:

input: $[s_1]$
output: $[s_1/P_1, s_1/P_2]$

The user asks for service s_1 and the recommender replies with a list of providers that offer s_1.

Query 2:

input: $[s_1]$ & $[s_1 (\sim/\mathcal{E})]$
output: $[s_1/P_1, s_1/P_2]$ & $[s_i/P_i]$

The user asks for service s_1 and/or a service similar to s_1. The recommender replies with a list of providers that offer s_1 and/or a list of providers who offer services similar with s_1.
"$\sim/\mathcal{E}$" represents the similarity of services with $\mathcal{E}$ as proximity

Query 3:

input: $[s]$ $[P_1, P_2]$
output: $[s_1/P_1, s_2/P_1]$ $[s_i/P_2, s_j/P_2]$

The user asks for a list of services offered by certain providers. The recommender replies with a list of services offered by those providers.

Query 4:

input: $[\, s \mid r > x]$
output: $[s_1/r_1, s_2/r_2]$

The user asks the recommender for a list of services, which has a ranking "r" higher than a certain value. The recommender replies with the list of services.

Some relevant queries P-R might be the following:

Query 5:

input: $[u_i]$
output: $[u_i\,[e/c]]$

The provider asks the recommender about user u_i. This may be relevant to the provider in order to assess the user's credibility. The recommender replies with the u_i expectation "e" and credibility "c".

Query 6:

input: [all U_i, $e > \alpha$, $c > \beta$]
output: [u_i [e/c]]

The provider asks the recommender for a list of users whose expectation and credibility are higher than a certain value. This may be relevant to the provider in order to assess the user's credibility. The recommender replies with the list of user(s).

Based on the formula presented in the following section, complex information can be gathered and more accurate answers to different queries can be provided.

3.3 Computation mechanism

The enhanced model allows a more comprehensive schema for computing the reputation.

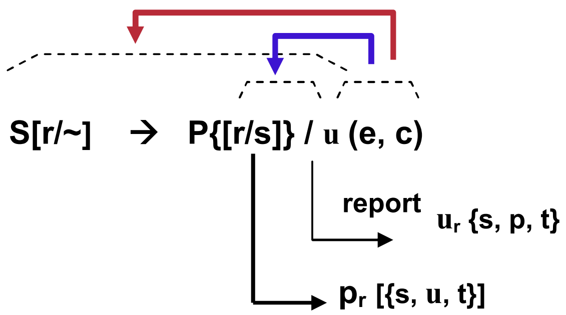

Figure 3. A computation schema for recommenders

In our framework, a recommender has mechanisms for representing services (S) with their reputation (r) and similarities (~), provider (P), with their reputation (r) linked to the reputation of their service providers (s), associated with user's (u) expectation (e) and credibility (c). A particular relation is valid at a moment (t). For example, a user x is expected to provide feedback with e = 80% and the confidence on its feedback is 70%. The feedback is on a provider (p) providing a service (s) at the time (t). The schema allows having a reputation view of a user at a given time, on a given provider delivering a given service. The schema also allows having a reputation of a provider, as perceived by a user at a given time, if delivered by a given service.

We are now going to concentrate on different scenarios dictated by the amount of data reported by users and service providers.

For example, a user sends a request to the Recommender for the best cell phone provider that would meet certain parameters. The Recommenders replies with provider P_1. The user makes a request for a number of cell phones from P_1. After the transaction is completed, all the involved parties have the option to send feedback to the Recommender. The Recommender collects the data and based on the feedback, it updates the reputation of the involved parties. The nature of the collected data can be divided in three main cases:

1. *Matching reports*

The number of feedback reports from the user matches the number of reports from the service provider within a particular time window relevant to the service type. To continue with the example from above, the user sends the feedback to the Recommender, including the number of cell phones that it purchased. P_1 reports to the Recommender that the user purchased a number of cell phones from it. The numbers reported by both the user and P_1 match.

A subclass of this scenario would be when P_1 sub-contracts from a different provider, P_2. If P_1 receives a request for cell phones, it can send the products from its own stock, send part from its own stock and part from P_2, or get the entire order from P_2. In this case, the Recommender would receive reports from both providers, P_1 and P_2. The exact number reported would not match since P_1 will report that it sent the entire order to the user, and P_2 would report that it sent a certain number of phones to P_1, but the data can be correlated. The correlation is done by using the transaction completion time, the user identifier, and the provider identifier.

2. *Over-reporting providers*

The number of feedback reports from user and provider does not match. This can be caused by either providers exaggerating the amount of transactions completed, or by users who underreport. In this case, some of the data can be correlated by the Recommender.

3. *Underreporting provider*

The number of feedback reports from user and provider does not match. This can be caused by either providers that do not report every transaction, or by users who exaggerate the amount of transactions completed. In this case, the Recommender can correlate some of the date.

4. Case study for reputation correction

Let us consider the following situation:

u -> [t] [p1] [s1], where u is the user, p1 and p2 are providers, t is the time of the request, and s1 is the service;
p1-> [t] [u] [s1], with p1 [r1/s1]
p2 → [t] [?][s1], with p2 [r2/s1]

and the following transaction reports:

|u|: reports α transactions
|p1| reports β transactions (with $\beta < \alpha$)
|p2| reports γ transactions

then

$k = (\beta - \gamma) / \alpha$

In this case, for a given user u, and for the considered service s1, the real reputation is $r_1' = k \times r_1$, as there is an indirect service delivery form p2 via p1 to the user u. The schema allows having a more accurate view on who is delivering a service.

Note: the number of transactions can be either reported or obtained by audit. In this use case, we consider that the providers are subscribed to an automated transaction report when delivering a service.

5. Discussion

In this section, we are comparing existing recommender systems with our proposal, on the basis of three main features: expectation, credibility, and user profile, as defined in Section 3.2.

5.1 Feature-based comparison

We consider a few well known recommender systems and only selected those three main features as a basis of comparison. The existing recommenders do not incorporate in the user profile the expectation and credibility of a given user.

Table 1. Feature based comparison of several recommender systems as well as the proposed one

	eBay	Amazon.com	Barnes & Nobles	proposal
expectation	Not in profile	Not in profile	Not in profile	Included in profile
credibility	Not in profile	Not in profile	Not in profile	Included in profile
User profile	yes	yes	yes	yes

While the considered systems (eBay, Amazon.com, Barnes & Nobles) make use of the notion of profile when recommending a product, the main target is to identify potential similar services and products to either satisfy a request or recommend a particular service unknown to the user (using the similarity concept).

By including these features, the recommender can have a more complete view on user's satisfaction based on more accurate information maintained by the system on the user's behavior (the degree of responsiveness of the user ability to give trusted rating).

5.2 Performance and accuracy

The performance and accuracy of a recommender system can be enhanced by including in the user's profile the user's expectancy and credibility. By having the expectancy of a user to leave a review and also its credibility, a recommender can better tune its suggestions to a user's requests with increased certainty. Ongoing experiments will identify the thresholds from where these features increase the accuracy of recommendations. Particular consideration will be given to the dynamics of user's feedback in terms of relationships between the frequency (volume) of the used services or products and the accuracy of the timely feedback.

6. Conclusion and future work

The paper presented a framework and appropriate mechanisms to evaluate the services/providers in the light of their respective direct impact on user perception. Essentially, the proposal considers several innovative ways of considering user impact on an accurate evaluation of a service/provider reputation. The proposed schema can capture indirect service delivery and allow reputation correction based on the real transactions.

Future investigations shod focus on a more formal definition of service/provider/feature similarity, and the stability of the reputation accuracy over a longer period. This might lead to the reputation predictions; specialized metrics for assessing the accuracy of predictions in the light of indirect delivery are challenging but seen as very helpful in web-service driven environment.

On the user side, consistency feedback and reliability should be correlated with the frequency of users' report and transaction peaks, as well as with the user's report patterns. This will allow detection of potential 'off-market' agreements between providers and set an appropriate service level agreement policy.

References

[1] P. Resnick and H. R. Varian. Recommender systems. Communications of the ACM, 40(3):56–58, 1997.

[2] Esma Aimeur and Flavien Serge Mani Onana. Better control on recommender systems. E-Commerce Technology, 2006. The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, The 3rd IEEE International Conference on Volume, Issue , 26-29 June 2006 pp. 38 – 38

[3] T. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12):61–70, 1992.

[4] Dieberger, A., Dourish, P., Hook, K., Resnick, P., and Wexelblat, A. Social navigation: techniques for building more usable system. Interactions, 7, 6, 2000, 36-45.

[5] Adomavicius, G. and Tuzhilin, A., Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, Volume 17, No. 6, June 2005. Pp. 734 – 749

[6] Chen, Y. B., and Xie, J. H. Online consumer reviews: a new element of marketing communications mix. Working paper, Eller College of Management, University of Arizona, Tucson, AZ, 2004.

[7] Kumar, N., and Benbasat, I. The influence of recommendations and consumer reviews on evaluations of Web sites. Information Systems Research, 17, 4, 2006, 425-439.

[8] Dini,O., Moh, M., Clemm, A.: Web Services: Self-adaptable Trust Mechanisms. AICT/SAPIR/ELETE 2005: 83-89

[9] Massa, P., & Bhattacharjee, B. (2004). Using Trust in Recommender Systems: An Experimental Analysis. Trust Management, (Proceedings of the 2nd International Conference, iTrust 2004).Oxford, UK, LNCS 2995, Springer, pp. 221-235.

[10] M. Pazzani, "A Framework for Collaborative, Content-Based, and Demographic Filtering, Artificial Intelligence Rev., pp. 393-408, Dec. 1999.

[11] A.I. Schein, A. Popescu, L.H. Ungar, and D.M. Pennock, Methods and Metrics for Cold-Start Recommendations," Proc. 25th Ann. Int'l ACM SIGIR Conf., 2002.

[12] R. Jurca and B. Faltings. "An Incentive Compatible Reputation Mechanism", Proc. IEEE Conf. on E-Commerce, pp. 285-292, Newport Beach, CA, June 2003.

[13] Pei-Yu Chen, Yen-Chun Chou, Kauffman, R.J. Community-Based Recommender Systems: Analyzing Business Models from a Systems Operator's Perspective. Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS-42). pp. 1-10, January 2009.

[14] Kwei-Jay Lin, Haiyin Lu, Tao Yu, and Chia-en Tai. A reputation and trust management broker framework for Web applications. Proceedings of The 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service. April 2005. pp. 262- 269

[15] Dellarocas, C. Strategic manipulation of Internet opinion forums: implications for consumers and firms. Mgmt. Sci., 52, 10, 2006, 1577-1593.