Personal-Hosting RESTful Web Services for Social Network Based Recommendation

Youliang Zhong, Weiliang Zhao, and Jian Yang

Department of Computing, Macquarie University,
North Ryde, NSW 2109, Australia
{youliang.zhong,weiliang.zhao,jian.yang}@mq.edu.au

Abstract. Recommender systems have been widely used in information filtering. However the existing recommendation methods do not work effectively in the situations when a group of people want to share information and make recommendations within a social network. In this paper we propose a personal-hosting web services architecture *ph-REST* for social network based recommendation, in which every user is represented by a dedicated RESTful web services engine that collaborates with others over a social structure formed by *co-peers with common interests.* The proposed architecture explores the potential of applying service and Cloud computing to personal and social information sharing and assimilation.

Keywords: Personal-hosting web services, Social network based recommendation, Recommendation methods.

1 Introduction

With the surge in the popularity of Web 2.0 technologies, people routinely use social networking, collaboration tools, and wikis to search, accumulate and acquire new knowledge, as well as to share the acquisitions with their friends and colleagues. Recommender systems have been widely accepted in information sharing and assimilation. However the current filtering methods have limitations in many situations where most items have few user ratings, and the users prefer to share their information within peer groups and make decisions as they wish.

Considering a scenario of research literature search, when a researcher wants to expand her collections in a particular area, she will ask her colleagues to give recommendations, and her colleagues may continue with the requests to their social associates and pass the results back to the researcher. This is usually a relay process for getting recommendations through social networks. To follow the human behaviour of social networking, it is necessary to build a truly distributed architecture with corresponding filtering methods by exploiting the 'social association' structure in recommendation process.

Web services [2] would be one of the promising technologies to achieve the above goal. Especially the recently popular RESTful web services architecture

G. Kappel, Z. Maamar, H.R. Motahari-Nezhad (Eds.): ICSOC 2011, LNCS 7084, pp. 661–668, 2011.

[6] has demonstrated its strengths in the applications where the services are consumed by a large number of clients simply through HTTP protocols.

In this paper we propose a personal-hosting web services architecture *ph-REST* with a relay-based recommendation method for social network based recommendation. In the architecture every user is represented by a dedicated RESTful web services engine referred to *REST-engine*, which collaborates with other users through web services provision and consumption. A recommendation process starts with a requesting user's collection of items and associated ratings. Then the user's friends or peers can forward the request to their friends or peers and so forth. All the peers produce and adjust recommendations, and return the recommendation results backwards to the requesting user. The main contributions of the paper are as follows:

– **An architecture of personal-hosting RESTful web services.** This paper proposes a personal-hosting RESTful web services architecture *ph-REST*, in which every user is equipped with a dedicated web services engine that plays both roles of service provider and service consumer. The proposed architecture shows a great potential of applying Service and Cloud computing to social network based recommendation.
– **A relay-based model for social network based recommendation.** This paper introduces a relay-based recommendation model by explicitly utilizing social association, which is based on a dynamically formed social structure. Having recommendations be adjusted by direct and indirect peers in the social structure broadens the range of recommending peers therefore increases the quality of recommendation results.

The rest of the paper is organized as follows. We firstly present a motivating example and the *ph-REST* structure in Section 2, then discuss the social network based recommendation model in Section 3. Prototype and experiments are discussed in Section 4, and the related work is reviewed in Section 5. Concluding remarks are provided in Section 6.

2 Personal-Hosting Web Services Architecture

2.1 A Motivating Example

Fig 1 illustrates how recommendations are produced followed "social association" in a social network by a relay mechanism. In the picture, each user maintains a collection of items and associated ratings. For instance, user *Alex* possesses a list of item "*a, c, e*", with associated ratings "*8, 7, 3*", he also has three friends: *John, Peter, Eddy*. *Alex* has commonly rated item *c* with *John*, and *e* with *Peter*, but nothing common with *Eddy*. We call an item like *c* or *e* as co-rated-item (CRI), and friends who have CRIs as co-peers. Consequently, *Alex* and *John* are co-peers, and so are *Alex* and *Peter*.

When user *Alex* wants to expand his collection. He sends a request to his co-peers *John* and *Peter*; similarly *John* can forward the request to his co-peer

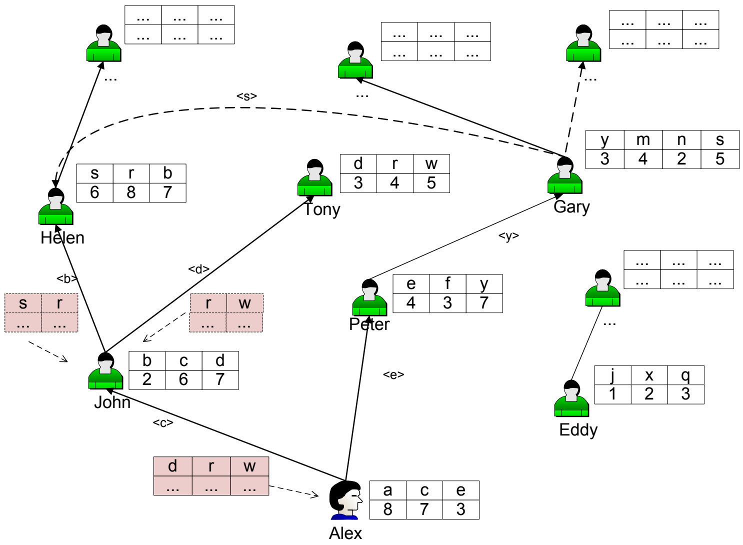


Fig. 1. A motivating example of social recommendation

Helen and *Tony*, and so forth. For a given request, the initial requestor and all its successive co-peers will form a social structure, and every user in the structure will produce recommendations on his own. Finally, after getting responses from his co-peers *John* and *Peter*, *Alex* aggregates and filters the recommended items to create a final recommendation list for himself.

2.2 ph-REST: Personal-Hosting RESTful Web Services

A personal-hosting web services architecture *ph-REST* is proposed, in which every user is represented by a dedicated web services engine (REST-engine). The communication between REST-engines follows a Producer-Consumer pattern [14]. Figure 2 shows the components of a REST-engine. Of each engine, there are *HTTP Server* and *HTTP Client* for communicating with other engines, *REST Router* handles recommendation requests and responses. In a *Recommendation application*, *Relay Manager* maintains co-peer relationship and communicates with recommendation components: *Prediction* and *Filtering*, which work with *Resource Manager* through to various data sources mapped to actual data in an embedded database.

3 Social Network Based Recommendation

3.1 Notation

From the motivating example, we introduce a co-peer graph CPG(q, V, A) as a labeled directed acyclic graph, where q is a recommendation request, V a set of vertices and A a set of directed arcs over V, such that,

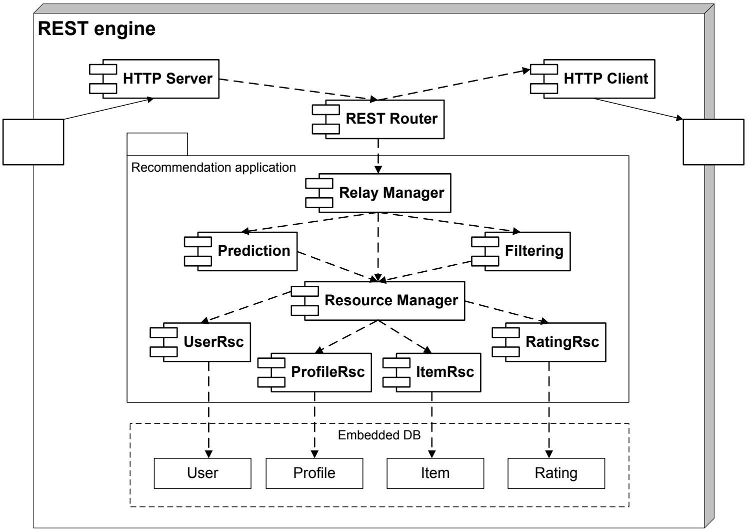


Fig. 2. The components of REST-engine

- For any $v \in V$, it represents a user that possesses a pair of tuples of items and ratings. $T(v) = \{t_i \mid i = 1..n\}$ is a set of items possessed by v, and $R(v) = \{v(t_i) \mid i = 1..n\}$ the associated ratings over $T(v)$.
- q is a recommendation request made by a root node $v_0 \in V$. The request q includes a user's preferences such as items and ratings. In Fig (1), *Alex* is the root node, he sends out a recommendation request with his possession of items (*a, c, e*) and ratings (*8, 7, 3*).
- For a pair of vertices $u, v \in V$, the set intersection of $T(u) \cap T(v)$ is denoted as C_u^v. If $C_u^v \neq \emptyset$, then C_u^v is referred to co-rated-items (CRI) between v and u, and u and v become co-peers. Consequently, a directed arc $a(u, v) \in A$ from u to v is established in the graph.
- For $v \in V$, a co-peer u that sends a request to v is called an inbound co-peer of v, or $Icp(v) = \{u\}$. And those co-peers that receive a request from v are called outbound co-peers of v, denoted as $Ocp(v)$. $Ocp(v) \cap Icp(v) = \emptyset$. Furthermore, of the outbound co-peers of v, a group of them that commonly rate on a particular item $t_i (t_i \notin T(v))$ is denoted as $Ocp_i(v)$.
- For $v \in V$, if $Ocp(v) = \emptyset$, then v is called a leaf peer in the graph. The set of all leaf peers is referred to as L.

3.2 Prediction Formulas

Every user in a co-peer graph may carry out three tasks when participating a recommendation process: (1) makes predictive ratings of potential items, which are rated by a user but not its inbound-co-peer. (2) aggregates the recommendations replied from its outbound co-peers, (3) selects highly recommended items based on predictive ratings or/and other criteria for its inbound co-peer.

Considering a pair of co-peers u and v, with co-rated-items C_u^v, and u is the inbound co-peer of v, or v is an outbound co-peer of u. For an item $t_i \in P_u^v$, where $P_u^v = T(v) \backslash T(u)$ indicating the "potential items" from v for u. We want to predict a rating for the item t_i, denoted as $r_u^v(t_i)$. Let us have $r_u^v(t_i) = v(t_i) + b$, where $v(t_i)$ is the rating made by v, and b an adjustment constant. According to Minimum Mean Square Error principle, we have $E = \sum_{t_j \in C_u^v}(v(t_j) + b - u(t_j))^2$, and the predictive $r_u^v(t_i)$ can be obtained by minimizing E. Formally,

$$\begin{gathered}
r_u^v(t_i) = v(t_i) + b, \\
b = \frac{1}{|C_u^v|} \sum_{t_j \in C_u^v}(u(t_j) - v(t_j)), \\
so,\ r_u^v(t_i) = \frac{1}{|C_u^v|} \sum_{t_j \in C_u^v}(v(t_i) - v(t_j)) + \overline{R}_v(u), \qquad (1) \\
where\ \overline{R}_v(u) = \frac{1}{|C_u^v|} \sum_{t_j \in C_u^v} u(t_j).
\end{gathered}$$

From the standing point of an inbound co-peer say user u, it may associate with multiple outbound co-peers which have made individual predictions on a same potential item say t_i. The user then needs to aggregate all the predictive ratings of t_i. To this end, we use the following formula (2) for simplicity of representation, where $r_u(t_i)$ is the aggregated rating of t_i for u.

$$\begin{gathered}
r_u(t_i) = \frac{1}{|Ocp_i(u)|} \sum_{v \in Ocp_i(u)} r_u^v(t_i). \qquad (2) \\
T(u) = T(u) \cup T^r(u),\ where\ T^r(u) = \cup_{v \in Ocp(u)} P_u^v, \\
R(u) = R(u) \cup \{r_u(t_j) \mid t_j \in T^r(u)\}.
\end{gathered}$$

3.3 Web Services Operations

In *ph-REST*, every user is represented by a REST-engine that provides three public operations: *init_request*, *relay_request* and *receive_recommendation*, and two internal methods: *wait_and_make* and *make_recommendation*. Both *init_request* and *relay_request* are used for receiving and relaying recommendation requests, and *receive_recommendation* for propagating recommendation responses. *wait_and_make* is designed to control recommendation process, and lastly *make_recommendation* to actually produce recommendations.

Through a full cycle of a replaying and adjusting process, a root node v_0 in a co-peer graph will get a set of recommended items and associated ratings $< T^r(v_0),\ R^r(v_0) >$, which will be aggregated and filtered by the root node.

$$T^r(v_0) = \cup_{u,v \in V}^{v \in Ocp(u)} \widetilde{P}_u^v,\ and\ R^r(v_0) = \{r_{v_0}(t_j) \mid t_j \in T^r(v_0)\}. \qquad (3)$$

4 Prototype and Evaluation

4.1 Prototype of ph-REST

A prototype system of *ph-REST* has been developed by using *Restlet*, one of the most popular RESTful web services framework. The experiment data were taken from ISI Web of Knowledge database [11], consisting of 500 articles, each bibliographic item of an article included title, author, publish year, abstract and keywords. 30 users were created with totally 635 ratings.

Besides the general considerations of web services deployment, several special issues have been addressed in the prototype system. Particularly, REST-engines need to control recommendation propagation under conditions. To this end, each request is relayed with control information including the endorsed co-peers, the maximum relay depth, and the longest period of waiting time. Doing so, the system ensures that the recommendation requests and responses are propagated through a well-controlled and tree-structured service chain.

4.2 Performance of Recommendation Model

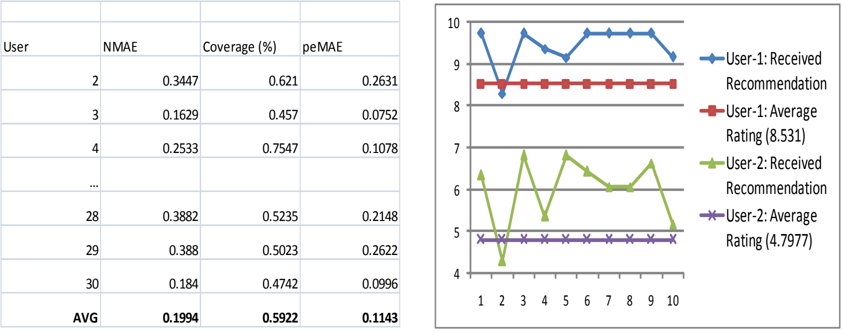
To measure the effectiveness of recommendation results, we adopt commonly used NMAE [7] and Coverage [3] metrics, as well as our own defined metric peMAE as follows (formula 4, 5 and 6). In these formulas, the superscript "e" stands for items or ratings originally made by a user, $r_v(.)$ for the predictive ratings made by co-peers. $T^r(v)$ represents the recommended items and $T_l^r(v)$ the highly relevant items from $T^r(v)$.

$$NMAE = \frac{1}{|V|} \sum_{v \in V} \frac{M_v}{R^e_{max}(v) - R^e_{min}(v)}, \quad M_v = \frac{1}{|T^e(v)|} \sum_{t_j \in T^e(v)} |r_v(t_j) - v^e(t_j)|. \quad (4)$$

$$Coverage = \frac{1}{|V|} \sum_{v \in V} \frac{|T_l^r(v)|}{|T^r(v)|}. \quad (5)$$

$$peMAE_v = \frac{1}{|T^r(v)|} \sum_{t_j \in T^r(v)} r_v(t_j) - \frac{1}{|T^e(v)|} \sum_{t_i \in T^e(v)} v^e(t_i). \quad (6)$$

Figure 3 shows our experiment results. The left table lists a set of NMAE, Coverage (%) and peMAE values. The average NMAE was about 0.2 and average peMAE marginally over 0.1, that indicates the recommendation results are fairly accurate. The average Coverage (%) was about 60%, that actually reflects the



User	NMAE	Coverage (%)	peMAE
2	0.3447	0.621	0.2631
3	0.1629	0.457	0.0752
4	0.2533	0.7547	0.1078
...			
28	0.3882	0.5235	0.2148
29	0.388	0.5023	0.2622
30	0.184	0.4742	0.0996
AVG	**0.1994**	**0.5922**	**0.1143**

a-Table: NMAE-Coverage-peMAE

b-Figure: Personalized recommendation

Fig. 3. Performance of recommendation relay model

nature of recommendation in social networks where novelty and serendipity usually get much more concerns than content relevance [9,10].

To exemplify how the proposed method achieves personalized recommendation, a set of commonly recommended items were selected from the recommendation results for two users. The average rating of user-1 was 8.5310 and that of user-2 was 4.7977. The right part of Figure 3 shows that these two users with different rating styles received individual recommendations that matched their personal modus operandi, even if the recommended items were the same.

5 Related Work

Generally speaking, two types of filtering approaches are widely used in existing recommender systems: content-based filtering (CN) and collaborative filtering (CF) [1]. While CN recommends items to a user based on the similarities between potential items and existing ones rated by the user [3,4], CF based on the ratings assigned by other users with "similar taste" [15,17]. Our prediction formulas leverage both *Slope One* method [12] and peer relationship so that recommendations can be relayed through the co-peers in social networks.

Few researches on recommendation methods deploy web services [19,20,16], all of them followed a centralized filtering approach. In contrast, our approach utilizes personal-hosting web services so that recommendations are produced and adjusted by every user in a social network.

While most researches on distributed recommendation focus on complexity, scalability and privacy-protection, few ones emphasize on distributed data sources or computation mechanism [18,5,8,13]. Nearly all these researches collect raw data from users, and may execute processing by using a global dataset. Our model does not rely on a global dataset nor a centralized process. Equipped with a personal-hosting web services engine, every user maintains its collection of data and produces recommendations on its own.

6 Conclusion

In this paper we propose a novel architecture *ph-REST* for social network based recommendation, in which every user is represented by a dedicated personal-hosting RESTful web services engine. The *ph-REST* architecture, the web services deployment and the prediction algorithms are discussed in the paper. The proposed architecture shows a great potential of using Service and Cloud computing in social network based recommendation.

Co-rated items and co-peer relationship are very basic relationships in social networking, there should be more meaningful factors that affect recommendation results, such as the depth of social relationship, the size of co-peers, and the centrality of peers. Future work can take these options into account during recommendation process, as well as the qualitative aspects of social networks such as trust and credibility.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the State-of-the-Art and possible extensions. IEEE Trans. on Knowl. and Data Eng. 17(6), 734–749 (2005)
2. Alonso, G.: Web services: concepts, architectures and applications. Springer, Heidelberg (2004)
3. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval, vol. 463. ACM press, New York (1999)
4. Belkin, N.J., Croft, W.B.: Information filtering and information retrieval: two sides of the same coin? Communications of the ACM 35(12), 29–38 (1992)
5. Berkovsky, S., Busetta, P., Eytani, Y., Kuflik, T., Ricci, F.: Collaborative Filtering over Distributed Environment. In: DASUM Workshop, Citeseer (2005)
6. Fielding, R.T.: Architectural styles and the design of network-based software architectures. PhD thesis, University of California, Irvine (2000)
7. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A Constant Time Collaborative Filtering Algorithm. Information Retrieval 4(2), 133–151 (2001)
8. Gong, S.J., Ye, H.W., Su, P.: A Peer-to-Peer based distributed collaborative filtering architecture. In: International Joint Conference On Artificial Intelligence, pp. 305–307 (2009)
9. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) 22(1), 5–53 (2004)
10. Iaquinta, L., de Gemmis, M., Lops, P., Semeraro, G., Filannino, M., Molino, P.: Introducing serendipity in a content-based recommender system. In: International Conference on Hybrid Intelligent Systems, pp. 168–173. IEEE (2008)
11. ISI-WoK (2010), http://wokinfo.com/
12. Lemire, D., Maclachlan, A.: Slope one predictors for online Rating-Based collaborative filtering. Society for Industrial Mathematics (2005)
13. Liu, Z., Qu, W., Li, H., Xie, C.: A hybrid collaborative filtering recommendation mechanism for p2p networks. Future Gener. Comput. Syst. 26, 1409–1417 (2010)
14. Milanovic, N.: Service engineering design patterns. In: International Workshop Service-Oriented System Engineering, pp. 19–26. IEEE (2006)
15. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: ACM Conference on Computer Supported Cooperative Work, pp. 175–186. ACM (1994)
16. Sen, S., Geyer, W., Muller, M., Moore, M., Brownholtz, B., Wilcox, E., Millen, D.R.: FeedMe: a collaborative alert filtering system. In: Anniversary Conference on Computer Supported Cooperative Work, pp. 89–98. ACM (2006)
17. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating word-of-mouth. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 210–217. ACM Press/Addison-Wesley Publishing Co. (1995)
18. Tveit, A.: Peer-to-peer based recommendations for mobile commerce. In: International Workshop on Mobile Commerce, pp. 26–29. ACM, Rome (2001)
19. Xu, J., Zhang, L.J., Lu, H., Li, Y.: The development and prospect of personalized TV program recommendation systems. In: International Symposium on Multimedia Software Engineering, pp. 82–89. IEEE (2002)
20. Zadel, M., Fujinaga, I.: Web Services for Music Information Retrieval (2004)