Recommendations Given from Socially-Connected People

Daniel González, Regina Motz, and Libertad Tansini

Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay
danielgonzalezbernal@gmail.com,{rmotz,libertad}@fing.edu.uy

Abstract. This paper presents how relationships among members of a social network can be used to explicitly specify the relevant features of a friendsourcing recommendation algorithm. One important contribution is to show how to conceptualize previous evaluations of items made by socially-connected users and the different features involved in this kind of algorithms, in a set of criteria for similarity between users in a social network. The paper presents how these specified criteria are used by the proposed friendsourcing recommendation algorithm and shows how the recommendation algorithm is integrated into a real recommender system to be used in a healthcare social network for the medical service of a university. Moreover, the work shows preliminary results which indicate that the information contained in social networks, processed with the proposed algorithm, is relevant for the generation of personalized recommendations.

Keywords: Friendsourcing, Recommendation Algorithm, Recommender Systems.

1 Introduction

According to the *Association for Computing Machinery* (ACM), recommender systems are computer applications which guide users in their decision-making during the time they interact with large amounts of information. Considering the explicit or implicit preferences expressed by users, recommender systems produce personalized recommendations of items [1]. There are various fields of application for these systems. For instance, they have a main role in electronic commerce (e-commerce), helping buyers to choose products and as a consequence, they substantially improve the sales [2]. Thus, the main goal of recommender systems is to find and retrieve information that satisfies the interests and needs of web users. The quality level of a recommender system is measured by the accurate and precise results it produces regarding the search intentions of the users [3]. When the context of recommendations is the web, the large amounts of available information as well as the massive usage of social networking websites imply big challenges for recommender systems, such as the ability to provide relevant results [3]. Considering the social networking, *Collaborative Filtering* (CF) is a well-known technique which improves recommender systems with a social feature [4, 5, 6]. In CF, items are recommended to new users based on the preferences of similar users [7]. CF, which falls in the category of

Y.T. Demey and H. Panetto (Eds.): OTM 2013 Workshops, LNCS 8186, pp. 649–655, 2013.

crowdsourcing techniques, uses the wisdom of crowd's theory [8]. In traditional *crowdsourcing*, problems are broadcast to an unknown group of people in a sort of an open call for solutions [3], whereas *friendsourcing* is a type of crowdsourcing aimed at collecting accurate information available only to a small, socially-connected group of people [9]. In line with this approach, there are several techniques to generate personalized recommendations. In this paper, we present mechanisms that collect accurate information available only from a trusted and eventually reduced, socially-connected group of people taking into account their friendship relations. This work also proposes what kind of information should be extracted from social networks and shows how the recommendation algorithm processes this information to achieve desirable results.

The remaining part of this paper is organized as follows: In Section 2, the friendsourcing-based recommendation algorithm is described. Section 3 discusses the integration of the algorithm into a real recommender system such as the one developed for the *Quality Health Information Retrieval* (QHIR) LACCIR project[1]. In Section 4, test cases and their preliminary results are described. Section 5 presents some conclusions and discusses improvements that could be made to the recommendation algorithm.

2 A Friendsourcing Recommendation Algorithm

The main contribution of this work is the proposal of a friendsourcing recommendation algorithm, which is thought to be used in a social network context, this implies that users are members of a social network and have relationships defined amongst them. Currently, there are many social networks with particular features or objectives, such as professional networks, celebrity news networks, for sharing videos and so on. In this way, it is desirable that the recommendation algorithm can be used independently of the kind of social network. In some cases, social networks bring the users the opportunity to choose the sort of relationship they share. For simplicity, the algorithm considers only one type of relationship between users in a social network, which is the *friendship*. Different types of relationships could be considered in future versions of the algorithm, but as first approach all the relationships are considered to be of the same sort. However, the opinion about an item given by a direct friend may not weight the same as the opinion about the same item given by a friend of a friend. Thus, the information contained in a social network can be modeled as an undirected graph, where the nodes are the users and the links are the relationships between them.

One input of the recommendation algorithm is the information contained in the social network, which is information about the users and their relationships. The algorithm takes the evaluations of items made by users as other input. Each user has the opportunity to rate the different quality attributes of the items. In this work, an item may be any object that can be identified and recommended to the users such as web pages, images, videos, and so on. Other input is the importance of each quality attribute. In this manner, the criteria of the recommendations are specified on the

[1] URL `http://www.laccir.org/web/#/Projects/HealthCare/InformationRetrieval/`

social network. Each user must have an identifier in order to produce personalized recommendations, so the algorithm takes the identifier of the user who will receive the recommendations, called *current user*, as other input of the algorithm. Additionally, a value which represents the maximum number of desired recommended items should be considered as another input. The algorithm returns a list of identifiers of recommended items ordered by the suggested importance for the current user.

Crowdsourcing is not used in this algorithm since it does not take into account the information available from the entire crowd to make recommendations. On the contrary, our algorithm takes the information available only from a trusted and eventually reduced, socially-connected group of people. One of the critical points of the algorithm is the selection of that group of people that will be used to make personalized recommendations. As previously mentioned, a social network can be modeled as an undirected graph, so we begin obtaining the users of the first level of the graph that has the current user as the root. In other words, the direct friends of the current user are obtained. Then, we order them based on similarity to the current user. The similarity between two users in a social network is defined as a normalized value between 0 and 1, which represents how similar they are, and is composed of three different components of similarity that will be described further on: the component of *similarity in the friendship relations*, the component of *similarity in the evaluations of items*, and the component of *similarity in the activity of rating items*. The first component to be taken into account, in order to find the similarity between users, is the similarity in the friendship relations. The more friends in common with the current user, the more similar they are. Let x be the current user and let y be a friend of x. Let $simFriend(x, y)$ be the component which represents the similarity in the friendship relations between two individuals x and y. We calculate its normalized value between 0 and 1, as shown in formula (1), where $nf(y)$ is the number of friends of the individual y, and $nfic(x, y)$ is the number of friends in common between the individuals x and y.

$$simFriend(x, y) = nfic(x, y) / (nf(y) - 1) \text{ if } nf(y) > 1, \text{ otherwise } simFriend(x, y) = 0 \quad (1)$$

Another component is the similarity in the evaluations of items, which captures to what extent two different users rate items in the same way. If both users rate the same items similarly in a positive or negative way, then they probably rate other items in a similar manner, that is, they have similar taste. Let $simEval(x, y)$ be the component which represents the similarity in the evaluations of items between the individuals x and y, we calculate its normalized value between 0 and 1, as shown in formula (2), where $qAttrEval(x, i, j)$ is the evaluation of item i produced by the individual x for the quality attribute j used in the recommendation criteria. The evaluations of the quality attributes of an item are normalized values between 0 and 1. In this work, the set of quality attributes $qAttr$ is composed of the following attributes: *trustworthy*, *objective*, *complete* and *well-written*. Let $niric(x, y)$ be the number of items rated in common between the individuals x and y, and let n be the number of quality attributes of the set $qAttr$.

$$simEval(x, y) = \left(\sum_{i=1}^{niric(x,y)} \left(\left(\sum_{j \text{ in } qAttr} (1 - |qAttrEval(x, i, j) - qAttrEval(y, i, j)|)\right) / n\right)\right) / niric(x, y) \text{ if } niric(x, y) > 0 \text{ and } n > 0, \text{ otherwise } simEval(x, y) = 0 \quad (2)$$

The last component is the similarity in the activity of rating items. It is important that members of the social network have an active attitude regarding the activity of rating items. A friend that has previously rated a large set of items is more trustworthy because his behaviour is better described by his ratings than other friends with less activity. Active friends have probably seen more items and hence give more trustworthy evaluations so it is important to consider this feature in the algorithm. If both users rate the same number of items, they probably have the same attitude considering the activity of rating items. Let *simActiv*(*x*, *y*) be the component which represents the similarity in the activity of rating items. We calculate its normalized value between 0 and 1, as expressed in formula (3) where *nir*(*x*) is the number of items rated by the individual *x*.

$$simActiv(x, y) = nir(y) / nir(x) \text{ if } nir(x) \geq nir(y) \text{ and } nir(x) > 0,\ simActiv(x, y) = nir(x) / nir(y) \text{ if } nir(y) > nir(x) \text{ and } nir(y) > 0, \text{ otherwise } simActiv(x, y) = 0 \quad (3)$$

A pondered value or weight, between 0 and 1, is defined for each of the three components of similarity. The main advantage of using weights is the possibility of making any component disappears by assigning zero to its weight, allowing more flexibility at the time of testing the algorithm and analysing the accuracy of the results. Let *sim*(*x*, *y*) be the similarity between the individuals *x* and *y*, and let α, β, γ be the respective weights for the three components. The sum of the three components is less or equal to 1. We calculate its normalized value between 0 and 1, as shown in formula (4).

$$sim(x, y) = \alpha * simFriend(x, y) + \beta * simEval(x, y) + \gamma * simActiv(x, y) \quad (4)$$

After the group of friends is ordered by similarity, the algorithm obtains the best rated items for each friend starting with the most similar friend of the current user. The rating for each item is a calculated value based on the evaluations of the quality attributes. Weights, which are values between 0 and 1, are also defined for each quality attribute so it is possible to set the importance for each one. The sum of the weights is less or equal to 1. We define a *value of acceptance* in order to require a minimum level of quality of the recommended items. Let *wqAttr*(*j*) be the weight defined for the quality attribute *j* of the set *qAttr*. Let *rating*(*f*, *i*) be the rating of the item *i* made by the individual *f*, friend of the current user. We calculate its value as shown in formula (5).

$$rating(f, i) = \sum_{j \text{ in } qAttr} (wqAttr(j) * qAttrEval(f, i, j)) \quad (5)$$

By ordering the items by its ratings, a preliminary list of recommendations is produced. After that, each item is reevaluated with the following process: if the rating is greater or equal to the value of acceptance, then the identifier of the item is added to the final ordered list of identifiers of recommended items. This process is repeated until the maximum number of recommended items is reached.

3 Integration of the Algorithm

In the QHIR LACCIR project, a healthcare social network for the Medical Service of the University of Cauca in Colombia and a recommender system, have been developed. The Medical Service provides care services and activities of disease preventions [10]. In order to integrate the recommendation algorithm into the recommender system we developed a set of classes in Java which implements the proposed algorithm, together with a main method and a web service which invokes that method. We represent the social network, the evaluations of items and the weights of the quality attributes by xml files.

Sample xml files can be accessed from the following website: `http://www.fing.edu.uy/inco/grupos/sis/www/index.php?content=./miembros/dgonzalez`.

To obtain the personalized recommendations, the main class *Friendsourcing* must be instantiated and the main method *friendsourcingalgorithm* must be invoked as shown in the following code:

```
Friendsourcing f = new Friendsourcing();
ArrayList<String> recommendations =
f.friendsourcingalgorithm(id, maxitems, pathsocialnet,
pathevaluations, pathweightsattr);
```

Where *id* is the identifier of the current user, *maxitems* is the maximum number of recommended items, *pathsocialnet* is the path of the social-network xml file, *pathevaluations* is the path of the evaluations-of-items xml file, *pathweightsattr* is the path of the weights-of-quality-attributes xml file, and *recommendations* is the list of identifiers of recommended items.

4 Test Cases and Preliminary Results

This section describes the test cases generated in order to test the recommendation algorithm. This section also explains the goal for each test case and analyzes the preliminary results. To evaluate the efficiency of the results obtained by the algorithm, we use two measures commonly used in the information retrieval, which are *precision* and *recall* [11]. Since the recommendation algorithm has three different components of similarity, we define three test cases to individually test each component. Four users of the social network are randomly chosen and recommendations of items are generated for each user using the algorithm. The selected users have a number of friends close to the average, which is considered a suitable selection criterion, as the tests aim to analyze the algorithm in general cases. Then, standard deviations for the measures of efficiency are also calculated. The tests are made in a social network with fifty users and ten items. The weights of the quality attributes have the same value (0.25), and the value of acceptance is set to 0.5.

The first test case aims to test the component of similarity in the friendship relations, so we set the weight for that component to the maximum value and eliminate the other two components, setting their value to zero. The results depicted in Table 1, shows that the evaluations of items made by friends that have a common circle of friends with the current user, should be considered when recommending items.

Table 1. Results of testing the component of similarity in the friendship relations

	Mean	Std. Deviation
Precision	0.650	0.167
Recall	0.718	0.282

The second test case aims to test the component of similarity in the evaluations of items, so we set its weight to the maximum value and eliminate the other two components in order to test individually that component. The results depicted in Table 2, shows that the similarity in the evaluations of items made by similar users is also useful in order to select items of interest to the current user.

Table 2. Results of testing the component of similarity in the evaluations of items

	Mean	Std. Deviation
Precision	0.700	0.224
Recall	0.719	0.180

The third test case aims to test the component of similarity in the activity of rating items, as in the other test cases, we set the weight for that component to the maximum value and eliminate the other two components. The results depicted in Table 3 shows that the similarity in the activity of rating items may be relevant for the generation of recommended items.

Table 3. Results of testing the component of similarity in the activity of rating items

	Mean	Std. Deviation
Precision	0.750	0.219
Recall	0.782	0.219

5 Conclusions and Future Work

In most cases, the recommendation algorithms used in websites are not made public due to the impact they have on certain activities on the web, such as e-commerce. Therefore, the knowledge of such algorithms is very valuable. Despite that fact, we designed a recommendation algorithm to be used in recommender systems which uses the information contained in the social network as well as evaluations of items made by socially-connected users. The recommendation algorithm was integrated into a real recommender system, specifically the one developed for the QHIR LACCIR project. Regarding the results of the performed tests, we may conclude that the similarity in the friendship relations should be considered when selecting similar users. The similarity in the evaluations of items made by similar users is also useful in order to select items of interest to the current user. Considering the similarity in the activity of rating items may be relevant to recommender systems, at the time of generating personalized recommendations. Extensions of the recommendation algorithm can be made as future work. For instance, the information contained in the profile of the members of the social network, such as demographic and academic information, may be incorporated to find similarity between users. Other future work includes generating larger test cases to further analyse the performance of the algorithm. Another interesting line of future work is evaluating its use in a real scenario such as the recommender system developed for the QHIR LACCIR project which is currently starting to be used by healthcare workers and needs to be in production for some time before it is possible to apply our algorithm, since there has to be at least a minimum number of users and evaluations of items.

Acknowledgements. This work was partially funded by: the LACCIR R1210LAC007 Project and PEDECIBA. We would especially like to thank the members of the QHIR LACCIR project for their collaboration in this work.

References

1. The ACM Conference Series on Recommender Systems, `http://recsys.acm.org`
2. Schafer, J.B., Konstan, J., Riedl, J.: Recommender Systems in E-Commerce. In: EC 1999: Proceedings of the 1st ACM Conference on Electronic Commerce, pp. 158–166. ACM, New York (1999)
3. Díaz, A., Motz, R., Fernández, A., Valdeni de Lima, J., López, D.: Quality Web Information Retrieval: Towards Improving Semantic Recommender Systems with Friendsourcing. VI Congreso Ibero-Americano de Telemática (2011)
4. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based Collaborative Filtering Recommendation Algorithms. In: WWW 2001: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295. ACM, New York (2001)
5. Bergman, L., Tuzhilin, A., Burke, R., Felfernig, A., Schmidt-Thieme, L. (eds.): RecSys 2009: Proceedings of the Third ACM Conference on Recommender Systems. ACM, New York (2009)
6. Jannach, D., Geyer, W., Dugan, C., Freyne, J., Singh Anand, S., Mobasher, B., Kobsa, A.: Workshop on Recommender Systems and the Social Web. In: RecSys 2009: Proceedings of the Third ACM Conference on Recommender Systems and the Social Web, pp. 421–422. ACM, New York (2009)
7. Good, N., Schafer, J.B., Konstan, J.A., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J.: Combining Collaborative Filtering with Personal Agents for Better Recommendations. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence, pp. 439–446 (1999)
8. Surowiecki, J.: The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. Doubleday (2004)
9. Bernstein, M., Tan, D., Smith, G., Czerwinski, M., Horvitz, E.: Personalization via Friendsourcing. ACM Transactions on Computer-Human Interaction (2010)
10. López, D., Blobel, B., González, C.: Architectural Approach for Quality and Safety Aware HealthCare Social Networks (2012)
11. Manning, C., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)