Producing Friend Recommendations in a Social Bookmarking System by Mining Users Content

Matteo Manca, Ludovico Boratto, Salvatore Carta
Dip.to di Matematica e Informatica
Università di Cagliari
Via Ospedale 72
09124 Cagliari, Italy
Email: {matteo.manca,ludovico.boratto,salvatore}@unica.it

***Abstract*—Social Bookmarking Systems (and Social Media Systems in general) are experiencing a quick growth in the number of active users. This expansion led to the well-known "social interaction overload" problem, that means that each user has too many potential people to interact with. In order to address this problem, user recommender systems are widely proposed in the social media literature to recommend friends or people to follow. Currently, there are no approaches able to produce friend recommendations in the Social Bookmarking Systems domain. In this paper we propose a friend recommendation algorithm for a Social Bookmarking System, based on low computational effort heuristics that allow real time applications. Experimental results show that, when users tag in the same way and are also interested in the same content, they can be recommended as friends. The proposed algorithm produces better results, with respect to policies that use only tags and do not consider content.**

Keywords*—*Social Bookmarking; Friend Recommendation; Tagging System.

I. INTRODUCTION

With the explosion of the Web 2.0, we observed a rapid growth of Social Bookmarking Systems and, in general, of all the forms of Social Media Systems. Social Bookmarking Systems allow users to use keywords (*tags*) to describe web pages that are of interest for them, helping to organize and share the resources with other users in the network [14]. The most widely-known example of Social Bookmarking Systems is Delicious.

In this domain, where users are connected and interact with each other, the growth of the population and the large amount of content led to scarcity of attention and to the well-known "social interaction overload" problem [7], [8]. These two problems are strongly related, since each user has too many potential users and items to interact with and this does not allow to focus on users or items that might be interesting for her/him. As a solution, the recommender systems research area recently put a lot of attention in the Social Media Systems domain, by developing a new class of systems named *social recommender systems* [6]. One of the most important areas social recommender systems focus on is *user recommendation*. User recommendation in the social domain aims at suggesting *friends* (i.e., recommendations are built for pairs of users that are likely to be interested in each other's content) or *people to follow* (i.e., recommendations are built for a user, to suggest users that might be interesting for her/him) [7].

These systems can be classified into three categories:

1) Systems based on the exploration of social graphs, that analyze the set of people related to the considered user, in order to produce recommendations. These systems recommend either the closest users in the graph, like friends of friends and followees of followees (the most famous example of this type of systems is Facebook [17]), or perform a random walk on the graph, in order to recommend the users that have the highest probability to be crossed (the main reference for this type of systems is Twitter [2]).
2) Systems based on the analysis of the interactions of the users with the content of the system (tags, likes, shares, posts on news, bookmarks, pictures, etc.). In order to exploit the interests, these systems usually apply complex algorithms. For example, some approaches build a user profile using TF-IDF (Term Frequency - Inverse Document Frequency) vectors [20] that, in order to be built, need to analyze each content the user interacts with [11]. Recommendations are produced by identifying users with similar profiles.
3) Hybrid systems, that consider both the social graph and the interactions of the users with the content (an example is represented by [9]). The use of different sources of data to produce the recommendations increases the complexity of these systems.

As highlighted in the previous classification, social recommender systems that recommend users are often based on approaches that filter content, make classifications and explore graphs. These systems certainly achieve a high accuracy but most of them are so complex that it would be hard to apply them to a real world scenario that, as previously said, grows quickly and involves huge amounts of data. The application of a complex algorithm to a real world scenario would involve difficulties in capturing the evolution of the users interests when building the recommendations.

Since user recommendation in a social domain aims at suggesting friends or people to follow, it is important to notice that the recommendation of a friend involves mutual interests and that the list of recommended *friends* might be different from the list of recommended *people to follow*. In fact, given two users u_i and u_j, u_j might be interesting for u_i, but not vice versa. This means that u_j would be recommended to u_i as a user to follow, but not as a friend.

 ISBN: 978-1-61208-311-7

So, the design of these two types of systems is different, since they involve different notions of users similarity. To the best of our knowledge, there are no approaches in literature that build friend recommendations in a Social Bookmarking System.

This paper presents a friend recommendation algorithm in a social bookmarking system that, by mining the content of the target user, recommends users that have similar interests. The algorithm has the capability to make a selective use of the available information and does not consider the social graph, in order to use as less information as possible. For this reason, it lends itself well to real time evaluations. The algorithm has been compared with two other reference algorithms, in order to evaluate the performances in terms of accuracy and to infer which aspects are more beneficial to produce recommendations in this domain; another aspect that we explored is the trade-off between the accuracy of the algorithm and the number of users involved in the recommendation process.

Our work brings relevant scientific contributions to the social recommender systems research area, now described in detail:

- This is the first algorithm able to recommend friends in a Social Bookmarking System.
- This algorithm is able to exploit the interests of a user in a selective way and produce recommendations using a simple approach, that can be applied in real time.
- The proposed algorithm has been tested, in order to evaluate how the considered information should be exploited (i.e., what information should be used and which weight should the considered interests have in the recommendation algorithm).

The proposed algorithm, both for its simplicity and because it is the first developed in this application domain, puts the basis on a research area previously not explored in the rapidly growing domain of social bookmarking systems.

The rest of the paper is organized as follows: Section II presents a formalization of a social bookmarking system; Section III describes the details of the recommender algorithm presented in this paper; Section IV illustrates the conducted experiments; Section V presents related work and Section VI contains comments, conclusions and future work.

II. Social Bookmarking Systems

Starting from the definition of a Social Tagging System given by Zhou et al. [16], we can state that a Social Bookmarking System consists of a set of users U, a set of bookmarks B, a set of tags T and a set of links between users L. Let $S = \{U, B, T, L\}$ be a Social Bookmarking System where:

- $U = \{u_i\}_{i=1}^{n}$ is a set of n users;
- $B = \{b_i\}_{i=1}^{w}$ is a set of w bookmarks;
- $T = \{t_i\}_{i=1}^{k}$ is a set of k tags;
- $L = \{l_i\}_{i=1}^{m}$ is a set of m links between pairs of users; these links may be bi-directional (i.e., a friendship) or uni-directional (i.e., one user follows the other).

Starting from the definition given above, we can define:

- $UB \subseteq B \times U = \{b_i | b_i \in B$ is a bookmark tagged by user $u \in U\}$ is the set of bookmarks used by u;
- $UT \subseteq T \times U = \{t_i | t_i$ is a tag used by user $u \in U$, $t_i \in T\}$ is the set of tags used by u;
- $BT \subseteq T \times U \times B = \{t_i | t_i$ is a tag used by user $u \in U$ to annotate the bookmark $b \in B$, $t_i \in T\}$;

The algorithm presented in this paper aims at finding previously unknown bi-directional links $l_k \in L(u_i, u_j)$, in order to recommend a friendship between user u_i and u_j.

III. Recommending Friends in a Social Bookmarking System

A. Motivation

The motivation of our algorithm is twofold. As mentioned in the Introduction, to the best of our knowledge there are no studies that propose an approach to recommend friends in the Social Bookmarking Systems domain. Secondly, a relevant aspect of a recommender system that operates in the social domain is the need to capture the user interests using lightweight algorithms; in fact too complex approaches may require too much time to infer the users interests. Therefore, when recommendations are produced, the estimated interests of a user may not consider the current ones that, in the meanwhile, may have been updated. So, motivated by the thesis proposed in [16] that the tagging activity of the users reflects their interests and by the intuition that users with similar interests use similar tags and the same bookmarks, we developed an algorithm that, given a Social Bookmarking System S, makes a selective use of the available information about interests to produce accurate friend recommendations. To be more precise, our algorithm computes user similarities with low computational cost metrics based on the set of bookmarks B and on the set of tags T.

B. Algorithm

Our algorithm is based on two similarity metrics, computed considering the tags and the bookmarks used by a user. Given a target user $u_t \in U$, the algorithm recommends the users with high tag-based and bookmark-based similarities. The algorithm works in three steps:

1) *Tag-based similarity computation.* The first similarity calculated among a target user u_t and the other users, is based on the tags used by each user. Given the number of times each tag was used by a user, Pearson's correlation is used to derive the similarity.
2) *Bookmark-based similarity computation.* The second type of similarity is the percentage of common bookmarks among u_t and the other users.
3) *Recommendations selection.* This step recommends to u_t the users with both a tag-based and a bookmark-based similarity higher than a threshold value.

In the following, we will give a detailed description of each step.

 ISBN: 978-1-61208-311-7

1) Tag-based Similarity Computation: Considering the previously given definition of $S = \{U, B, T, L\}$, we represent each user u with a vector $\overrightarrow{v_u} = \{v_{u1}, v_{u2}, ..., v_{uk}\}$, where each element v_{ui} is the relative frequency of each tag $t_i \in T$ used by $u \in U$ and is computed as follows:

$$v_{ui} = \frac{f_{ui}}{\#UT(u)} \tag{1}$$

Value f_{ui} represents the frequency of a tag $t_i \in T$ for user u. Given that each user is represented by a vector based on tag frequencies and that [16] states that users' interests are reflected in their tagging activities, our algorithm computes the first user similarity with the Pearson's correlation coefficient [19], to infer users with similar interests. We chose to use this metric because, as proved by Breese et al. [15], it is the most effective technique for the similarity assessment among users.

Let $\{u, m\}$ be a pair of users represented respectively by vectors $\overrightarrow{v_u}$ and $\overrightarrow{v_m}$. Our algorithm computes the tag-based user similarity tb as defined in (2):

$$tb(u,m) = \frac{\sum_{i \subset T_{um}} (v_{ui} - \overline{v}_u)(v_{mi} - \overline{v}_m)}{\sqrt{\sum_{i \subset T_{um}} (v_{ui} - \overline{v}_u)^2} \sqrt{\sum_{i \subset T_{um}} (v_{mi} - \overline{v}_m)^2}} \tag{2}$$

where T_{um} represents the set of tags used by both users u and m and values $\overline{v}_u$ and $\overline{v}_m$ represent, respectively, the mean of the frequencies of user u and user m. The metric compares the frequencies of all the tags used by the considered users. The similarity values range from 1.0, that indicates complete similarity, to -1.0, that indicates complete dissimilarity. Herlocker et al. [13] demonstrated that negative similarities are not significant to evaluate the correlation among users, so in our algorithm we consider only positive values.

2) Bookmark-based similarity computation: To increment the system knowledge on user interests, our algorithm combines the tag-based similarity presented above with another metric based on bookmarks. The metric calculates the percentage of common bookmarks between two users u and m.

Let us consider $UB(u)$, i.e., the set of bookmarks used by a user $u \in U$. We define $D(u, m) = UB(u) \cap UB(m) = \{b_i | b_i \in UB(u) \wedge b_i \in UB(m)\}$ as the sets of bookmarks used by both user u and user m. Given a pair of users $\{u, m\}$, we compute the bookmark-based user similarity *bb*, by considering the common bookmarks among the users, as follows:

$$bb(u,m) = \frac{\#D(u,m)}{\#UB(u)} \tag{3}$$

where $\#D(u, m)$ and $\#UB(u)$ represent, respectively, the cardinality of the sets $D(u, m)$ and $UB(u)$. We can notice that, since the bb metric is calculated as a percentage, the similarity is based on the number of bookmarks used by the user that we are comparing (i.e., $\#UB(u)$). This means that, differently from previously computed metric, similarity $bb(u, m)$ can be (and often it is) different from $bb(m, u)$. Our algorithm considers both values.

3) Recommendations selection: Once the tag-based and the bookmark-based user similarities are computed for each pair of users, our algorithm chooses a set of users to recommend to the target user by selecting:

- the ones that have a tag-based user similarity higher than a threshold value α (i.e., $tb > \alpha$);
- the ones that have a bookmark-based user similarity (at least one of the two computed) higher than a threshold value β (i.e., $bb > \beta$).

So, given a target user u_t, the candidate set $CS(u_t)$ of users to recommend is selected as follows:

$$CS(u_t) = \{u_i \in U \,|\, tb(u_t, u_i) > \alpha \,\&\&(bb(u_t, u_i) > \beta) \,\|\, (bb(u_i, u_t) > \beta)\} \tag{4}$$

IV. EXPERIMENTAL FRAMEWORK

This section presents the framework used to perform the experiments. The dataset used and the data preprocessing are first described. Then, the metrics used for the evaluation are presented. The last part of the section presents the experimental setup and the obtained results.

A. Dataset and pre-processing

Experiments were conducted on a Delicious dataset distributed for the HetRec 2011 workshop [12]. It contains 1867 users, 69226 URLs, 7668 bi-directional user relations, 53388 tags, 437593 tag assignments (i.e., tuples [user, tag, URL]), 104799 bookmarks (i.e., distinct pairs [user, URL]).

We pre-processed the dataset, in order to remove all the users that were considered as "inactive", i.e., the ones that used less than 5 tags and less then 5 URLs.

B. Metrics

Given a set of recommendations $R = \{\cup CS(u_t), \forall u_t \in U\}$ and a set of correct recommendations $C \subseteq R$ (i.e., the pairs of recommended users that also appear in the dataset as a bi-directional user relations), recommendation *accuracy* is defined as the ratio of correct recommendations among all recommendations and it is computed as showed in (5).

$$accuracy = \frac{\#C}{\#R} \tag{5}$$

The other aspect considered in the evaluation is the *user coverage*, that represents the percentage of users involved in the recommendations, i.e., for how many users the algorithm is able to produce recommendations, given a specific threshold value. The metric can be computed as follows:

$$userCoverage = \frac{\#R}{\#U} \tag{6}$$

C. Experiments

Strategy. We performed two different experiments. The first aims to make an *evaluation of the recommendations*, by exploring the accuracy of the algorithm with different thresholds, while the other aims to make an *evaluation of the user coverage*, by exploring the trade-off between accuracy and user coverage.

In order to evaluate the recommendations we implemented a state-of-the-art policy [16], that we used as reference algorithm. Zhou et al. [16] developed a tag-based user recommendation framework and demonstrated that tags are the most effective source of information to produce recommendations. We compare the performances of our algorithm with respect to that of the reference one, that uses only tags i.e., with $bb = 0$), in terms of accuracy. Supported by the thesis that the use of only one source of data leads to better performances, we designed a second reference algorithm that considers only the bookmark-based similarity (i.e., with $tb = 0$).

In order to explore the trade-off between the accuracy analyzed in the previous experiments and the user coverage, we evaluate how the number of involved users (i.e., the user coverage) changes with respect to the tag-based user similarity tb and the bookamrk-based user similarity bb.

During the analysis of the accuracy, we evaluated all the values of parameters α and β between 0 and 1, using a 0.1 interval. When analyzing the user coverage, we also considered the values of β between 0 and 0.1, with a 0.01 interval, in order to evaluate in more the detail the user coverage studied considering bb (results will help motivating our choice to extend this analysis).

The experimental setup and the results are now described.

Evaluation of the recommendations. Given a target user u_t, the algorithm built a candidate set, $CS(u_t)$, of users to recommend. For each recommended user $u_i \in CS(u_t)$, we analyze the bi-directional user relations in the dataset to check if there is a link between the target user u_t and the recommended user u_i (i.e., if the users are friends). Let $R = \{\cup CS(u_t), u_t \in U\}$ be the set of all recommendations. We consider as *correct recommendations* the set $C \subseteq R$ of all the recommendations for which there is a correspondence in the relations of the dataset. This experiment analyzes the amount of correct recommendations in terms of *accuracy*. Given different values of α and β, the accuracy of the algorithm is calculated, in order to analyze how the performances of the algorithm vary as the similarities between users grow. The obtained results are illustrated in Fig. 1 and Fig. 2.

Fig. 1 shows how the accuracy values change with respect to the bookmark-based user similarity bb. The figure contains a line for each possible value α of the tag-based user similarity tb. We can observe that the accuracy values grow proportionally to the bb values. This means that the more similar are the users (both in terms of tag-based similarity and of bookmark-based similarity), the better the algorithm performs. However, for bb values higher than 0.5 no user respects the constraints, so we cannot make any recommendation; this is the reason why there are no accuracy values for bookmark-based user similarities higher than 0.5 ($bb > 0.5$). Fig. 2 shows the same results from the tag-based user similarity point of view. The

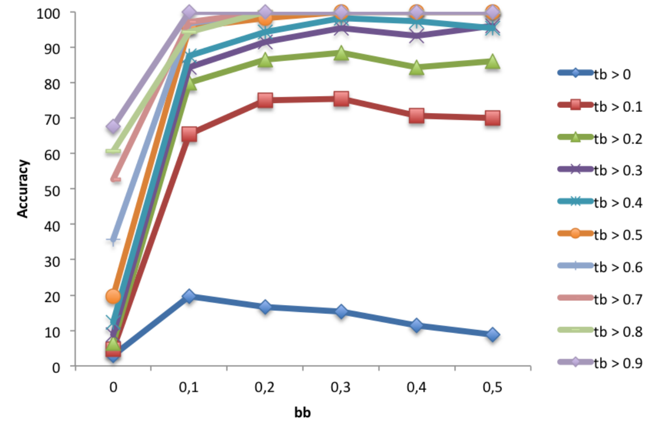

Fig. 1. Accuracy of the algorithm with respect to bookmark-based user similarity bb, for each value of the tb user similarity

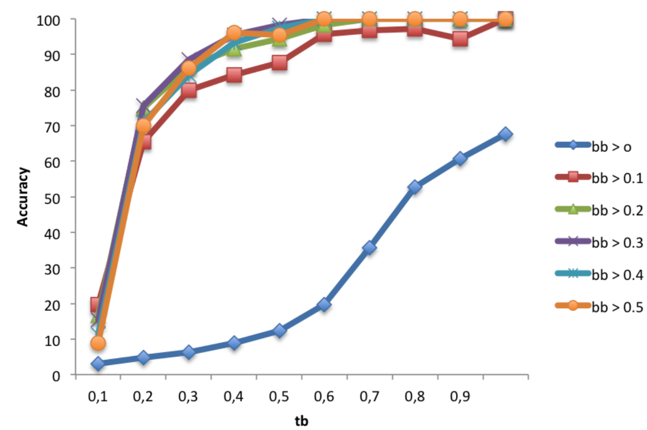

Fig. 2. Accuracy of the algorithm with respect to tag-based user similarity tb, for each value of the bb user similarity

figure illustrates the accuracy values with respect to the tag-based user similarity tb; here each line presents the results for a given value β of the bookmark-based user similarity bb. As results show, also from this perspective, the accuracy values grow proportionally to the tb values. The red lines in Fig. 1 and Fig. 2 show the results of the reference algorithms, where $tb = 0$ and $bb = 0$. In both cases, the two metrics combined improve the quality of the recommendations with respect to the cases where only one is used.

Evaluation of the user coverage. In this experiment, we study how the *user coverage* of the algorithm (i.e., the percentage of users involved in the recommendations) changes with respect to the tag-based user similarity tb and the bookmark-based user similarity bb. As Fig. 1 shows, when the behavior of the user coverage with respect to the bookmark-based user similarity bb is analyzed, each value of bb is combined with several values of tag-based user similarity tb. In this experiment we are interested only in the tb value that leads to the maximum values of user coverage. In the same way, we evaluate the user coverage with respect to the tag-based user similarity tb values, considering only the bookmark-based user

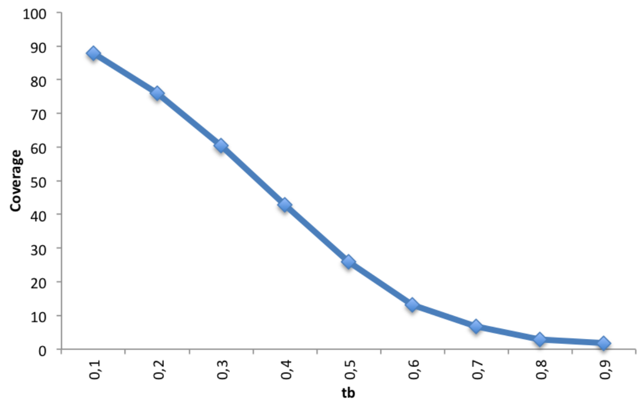

Fig. 3. User coverage of the algorithm with respect to tag-based user similarity tb

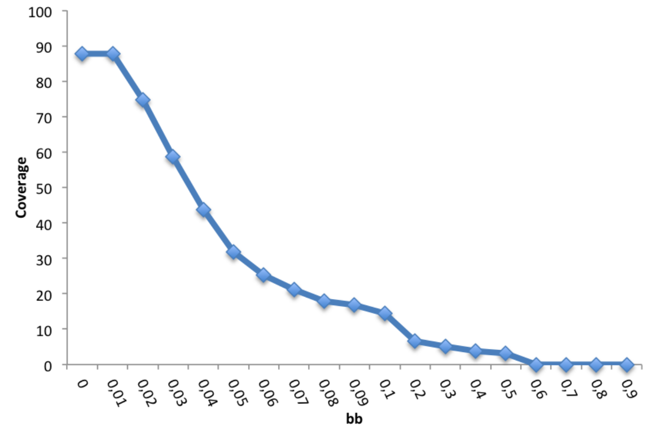

Fig. 4. User coverage of the algorithm with respect to bookmark-based user similarity bb

similarity bb that leads to the maximum user coverage values. Experiments are repeated with different values of α and β.

The results are presented in Fig. 3 and Fig. 4.

Fig. 3 shows the user coverage values with respect to the values of the tag-based user similarity tb; as previously mentioned, in this figure we do not have a line for each value of bookmark-based user similarity bb but we represent just a line that corresponds to $bb = 0.01$, which is the case that leads to the maximum values of user coverage. The same consideration can be made for Fig. 4, that represents the trend of the user coverage with respect to the bookmark-based user similarity bb; also in this case, we do not represent a line for each possible value of the tag-based user similarity tb, but just the values that correspond to $tb = 0.1$ (i.e., the value that allows to reach the maximum user coverage). As expected, high values of the thresholds α and β (that indicate a high similarity among users) correspond to low user coverage values. Effectively, in both Fig. 3 and Fig. 4 we can observe that we have user coverage values lower than 50% (that on a scale which ranges from 0 to 100 can be considered low values) for values of tb higher than 0.3 and for values of bb higher than 0.03. In Fig. 4 we can also observe that for values of the bookmark-based similarity bb higher than 0.5 the user coverage is 0 and that a consistent variation of the user coverage is between 0 and 0.1 (this is why we chose to extend our analysis by considering also those values).

V. Related Work

In the last years, Social Bookmarking Systems have been studied from different points of view. This section presents related work on user recommendation in this research area.

In [2], Gupta et al. present Twitter's user recommendation service based on shared interests, common connections, and other related factors. The proposed system builds a graph in which the vertices represent users and the directed edges represent the "follow" relationship; this graph is processed with an open source in-memory graph processing engine called Cassovary. Finally, recommendations are built by means of a user recommendation algorithm for directed graphs, based on SALSA (Stochastic Approach for Link-Structure Analysis). Our algorithm differs because we make friend recommendations and, furthermore, our algorithm uses just a restricted set of available information, without considering the social graph.

In [11], Chen et al. describe a people recommender system in an enterprise social network domain. They compare four algorithms, two based on social relationship information and two based on content similarity and demonstrate that the algorithms that use social information are stronger to find known contacts, while algorithms based on content similarities are better to discover new friends. We cannot compare with this approach, since it is applied to a delimited enterprise social network domain.

Guy et al. [10] describe a people recommender system for the IBM Fringe social network. The system uses enterprise information like org chart relationships, paper and patent co-authorship and project co-membership, which are specific of this social network, so it is hard to compare to them.

Hannon et al. [9] describe a followee recommender system for Twitter based on tweets and relationships of their Twitter social graphs. By using this information, they build user profiles and demonstrate how these profiles can be used to produce recommendations. In our work, we do not use any social connection information and furthermore we recommend friendship relationship and not users to follow.

In [3], a recommender system based on collocation (i.e., the position of the user) is presented. It uses short-range technologies of mobile phones, to infer the collocation and other correlated information that are the base for the recommendations. In our domain we do not have such a type of information, so we cannot compare with this algorithm.

Zhou et al. [16] propose a framework for users' interest modeling and interest-based user recommendation (meant as people to follow and not as a friend), tested on the Yahoo! Delicious dataset. Recommendations are produced by analyzing the network and fans properties. Differently from this framework, our algorithm produces friend recommendations.

In [1], a study about what cues in a user's profile, behavior, and network might be most effective in recommending people, is presented. As previously highlighted, we are interested in producing recommendations only based on users' content.

Liben-Nowell and Kleinberg [5] studied the user recommendation problem as a link prediction problem. They develop several approaches, based on metrics that analyze the proximity of nodes in a social network, to infer the probability of new connections among users. Experiments show that the network topology is a good tool to predict future interactions. We aim at using more basic information and not graphs or network topologies.

In [18], Arru et al. propose a user recommender system for Twitter, based on signal processing techniques. The considered approach defines a pattern-based similarity function among users and makes use of a time dimension in the representation of the users profile. Our algorithm is different because we aim at suggesting friends while on Twitter there is no notion of "friend" but it works with "people to follow".

VI. Conclusions

This paper presented a friend recommendation algorithm in the Social Bookmarking System domain as a means to link users with similar interests. The goal was to infer users' interests from content, making a selective use of the available information and without using complex algorithms, hard to apply to a real world scenario. As results show, our algorithm produces accurate recommendations by using the tags and the bookmarks used by users. We also explored the trade-off between recommendation accuracy and user coverage and observed that high values of similarity lead to low values of coverage. A comparison with a state-of-the-art policy, that considers only the tags, shows that the combined use of tags and bookmarks leads to improvements with respect to this one.

Future work will focus on evaluating the accuracy of the recommendations by using different metrics, like Precision and Recall, that allow both to measure the amount of correct recommendations and to evaluate the proposed algorithm from new perspectives.

Acknowledgment

This work is partially funded by Regione Sardegna under project CGM (Coarse Grain Recommendation) through Pacchetto Integrato di Agevolazione (PIA) 2008 "Industria Artigianato e Servizi".

References

[1] M. J. Brzozowski and D. M. Romero, "Who should i follow? recommending people in directed social networks," in *ICWSM*, 2011.

[2] J. L. Pankaj Gupta, Ashish Goel, A. Sharma, D. Wang, and R. Zadeh, "Wtf: The who to follow service at twitter," in *Proceedings of www2013 Conference*, 2013.

[3] D. Quercia and L. Capra, "Friendsensing: recommending friends using mobile phones," in *Proceedings of the third ACM conference on Recommender systems*, ser. RecSys '09. New York, NY, USA: ACM, 2009, pp. 273-276.

[4] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi, "Short and tweet: experiments on recommending content from information streams," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1185–1194.

[5] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proceedings of the twelfth international conference on Information and knowledge management*, ser. CIKM '03. New York, NY, USA: ACM, 2003, pp. 556–559.

[6] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Berlin: Springer, 2011, pp. 1–35.

[7] I. Guy, L. Chen, and M. X. Zhou, "Introduction to the special section on social recommender systems," *ACM TIST*, vol. 4, no. 1, p. 7, 2013.

[8] H. A. Simon, "Designing organizations for an information rich world," in *Computers, communications, and the public interest*, M. Greenberger, Ed. Baltimore: Johns Hopkins Press, 1971, pp. 37–72.

[9] J. Hannon, M. Bennett, and B. Smyth, "Recommending twitter users to follow using content and collaborative filtering approaches," in *Proceedings of the fourth ACM conference on Recommender systems*, ser. RecSys '10. New York, NY, USA: ACM, 2010, pp. 199–206.

[10] I. Guy, I. Ronen, and E. Wilcox, "Do you know?: recommending people to invite into your social network," in *Proceedings of the 14th international conference on Intelligent user interfaces*, ser. IUI '09. New York, NY, USA: ACM, 2009, pp. 77–86.

[11] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy, "Make new friends, but keep the old: recommending people on social networking sites," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 201–210.

[12] I. Cantador, P. Brusilovsky, and T. Kuflik, "Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011)," in *Proceedings of the fifth ACM conference on Recommender systems*, ser. RecSys '11. New York, NY, USA: ACM, 2011, pp. 387–388.

[13] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Research and Development in Information Retrieval*, American Association of Computing Machinery. American Association of Computing Machinery, 8/1999 1999.

[14] U. Farooq, T. G. Kannampallil, Y. Song, C. H. Ganoe, J. M. Carroll, and L. Giles, "Evaluating tagging behavior in social bookmarking systems: metrics and design heuristics," in *Proceedings of the 2007 international ACM conference on Supporting group work*, ser. GROUP '07. New York, NY, USA: ACM, 2007, pp. 351–360.

[15] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, ser. UAI'98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.

[16] T. C. Zhou, H. Ma, M. R. Lyu, and I. King, "Userrec: A user recommendation framework in social tagging systems," in *AAAI*, M. Fox and D. Poole, Eds. AAAI Press, 2010.

[17] F. Ratiu, "Facebook:people you may know," May 2008. [Online]. Available: https://blog.facebook.com/blog.php?post=15610312130

[18] G. Arru, D. F. Gurini, F. Gasparetti, A. Micarelli, and G. Sansonetti, "Signal-based user recommendation on twitter," in *WWW (Companion Volume)*, L. Carr, A. H. F. Laender, B. F. Lóscio, I. King, M. Fontoura, D. Vrandecic, L. Aroyo, J. P. M. de Oliveira, F. Lima, and E. Wilde, Eds. International World Wide Web Conferences Steering Committee / ACM, 2013, pp. 941–944.

[19] K. Pearson, "Mathematical contributions to the theory of evolution. iii. regression, heredity and panmixia," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Math. or Phys. Character (1896-1934)*, vol. 187, pp. 253–318, Jan. 1896.

[20] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.