

Accuracy Versus Novelty and Diversity in Recommender Systems: A Nonuniform Random Walk Approach

Georgios Alexandridis, Georgios Siolas and Andreas Stafylopatis

Abstract In this chapter, we focus on recommender systems that are enhanced with social information in the form of trust statements between their users. The trust information may be processed in a number of ways, including the random walks in the social graph, where every step in the walk is chosen almost uniformly at random from the available choices. Although this strategy yields satisfactory results in terms of the novelty and the diversity of the produced recommendations, it exhibits poor accuracy because it does not fully exploit the similarity information among users and items. Our work tries to model user-to-user and user-to-item relation as a probability distribution using a novel approach based on Rejection Sampling in order to decide its next step (biased random walk). Some initial results on reference datasets indicate that a satisfying trade-off among accuracy, novelty, and diversity is achieved.

Keywords Recommender systems · Trust networks · Non-uniform random walks · Rejection sampling · Accuracy · Novelty · Diversity

1 Introduction

It is a fact that the emergence of Online Social Networks (OSN) has altered our everyday experience with the Internet and the World Wide Web. A number of new application domains have been born, while others, traditional ones, have been enriched. The latter is the case with the recommender systems (RS), where OSN have leveraged user experience by allowing a more thorough interaction that surpasses the traditional user-to-item review.

G. Alexandridis (✉) · G. Siolas · A. Stafylopatis
School of Electrical and Computer Engineering,
National Technical University of Athens, 157 80 Zografou, Athens, Greece
e-mail: gealexandri@islab.ntua.gr

G. Siolas
e-mail: gsiolas@islab.ntua.gr

A. Stafylopatis
e-mail: andreas@cs.ntua.gr

Indeed, research in the traditional RS field had come to a relative standstill prior to the advent of social networks. Although the application of the latter into the former is still novice, current state-of-the-art research in the field involves the integration of OSN in one or another form [1, 2]. It could be further argued that the blending of the two areas has brought about a new research field, that of the socially aware recommendation.

Social recommender systems (SRS) model and exploit user-to-item and user-to-user interaction in a plethora of ways [3, 4]. The addition of social information generally leads to more novel and diverse recommendations. However, this does not necessarily imply that the recommendations would be accurate altogether; indeed SRS have to be selective in the volume of information they incorporate. In this context, random walks on the social graph are fit for this purpose, since they focus on those subsets of the data that they find useful. For this reason, almost from the very beginning, they have been a natural choice for researchers in the field and they have been used in the implementation of widely used and successful systems [5, 6]. It is not only our belief, but also that of the community [7, 8] that random walks have not yet revealed their full potential and that there is still room both for improvements in existing algorithms and the exploitation of other aspects of the random walks that are currently unexplored. In continuation of a preliminary work [9], this work tries to exploit the random walks from a different viewpoint; that of bridging the gap between recommendation accuracy on the one hand and novelty and diversity on the other.

2 Social Recommender Systems Based on Trust

Traditional RS can be extended by incorporating the interaction among users into them. This interaction may take place in a number of ways, the most common of which is Trust. It is the most simple form of user relation, where a user expresses his opinion on another user's behavior. Trust statements could either be binary (i.e., trust/distrust) or they may range over a broader set of values (usually in the $[0, 1]$ interval). It should be noted that trust does not necessarily imply correlation in the rating behavior [10].

The public release of socially enhanced recommendation datasets, such as the Filmtrust or the Epinions datasets (Table 1) has spurred interest in SRS. Since most of these datasets disclose trust information among their users, a substantial amount of the work in the area has evolved around trust-aware RS.

2.1 Trust Aggregation

A common way of processing the trust information of SRS is by aggregation; that is, to try to build a metric that would accumulate the available trust statements in the system. An obvious choice would be to consider all the paths that end up to a

Table 1 Recommender datasets used in the experiments

	Filmtrust	Epinions
Users	1919	49290
Items	2018	139738
Ratings	33526	664824
Ratings' density	1.15 %	0.01 %
Trust statements	1591	487182
Global clustering coefficient	0.0004	0.0002

particular user, in an effort to estimate his or her importance. Such SRS are also called *Reputation Systems* and their operation bears resemblance to the way the *PageRank* scoring algorithm works [7]. Although important research has been conducted in this direction, global trust metrics are not particularly suitable for the recommendation task. The main reason is that recommendations have to be personalized and in that sense the reputation of each user could not be constant; it depends on the viewpoint of each other user.

Local trust metrics, on the other hand, put the emphasis on each individual user and depart from him/her in order to explore the network. One of the earliest works in the field include the *gradual trust metric MoleTrust* [10] proposed by Massa and Avesani. The graph is first transformed into an acyclic form (a tree) by removing all loops in it and then the trust statements are accumulated in a depth-first fashion, starting from each user, up to each and every other user in the network. The *propagation horizon* determines the length of the exploration; the most common forms being MoleTrust-1, where only the users that target user trusts are considered, and MoleTrust-2, where the exploration also includes those trusted by those the target user trusts. If T_{u_t} is the set that includes all users in u_t 's network that have rated item i_t (which has not been evaluated by u_t yet), then the recommendation value $\widehat{r_{u_t, i_t}}$ is approximated using the following formula (*trust-based collaborative filtering*) :

$$\widehat{r_{u_t, i_t}} = \overline{r_{u_t}} + \frac{\sum_{u \in T_{u_t}} t_{u_t, u} (r_{u, i_t} - \overline{r_u})}{\sum_{u \in T_{u_t}} t_{u_t, u}} \quad (1)$$

where $\overline{r_{u_t}}$ is the mean of the ratings u_t has provided so far and $t_{u_t, u}$ the amount of trust u_t places on u .

Another popular gradual trust metric, proposed by Golbeck, is *TidalTrust* [6]. TidalTrust is different from MoleTrust in the sense that no propagation horizon is required for the accumulation of trust; instead the shortest path from the target user to each other user in the network is computed. All paths above a predefined threshold form the *Web of Trust (WOT)* for that particular user. If there exists more than one path between two users, then the one with the biggest value is chosen. If WOT_{u_t} is the set that includes those users in u_t 's web of trust network that have rated item i_t , then the recommendation value $\widehat{r_{u_t, i_t}}$ is approximated using the formula (*trust-based*

weighted mean):

$$\widehat{r_{u_i, i_t}} = \frac{\sum_{u \in T_{u_i}} t_{u_i, u} r_{u, i_t}}{\sum_{u \in T_{u_i}} t_{u_i, u}} \quad (2)$$

2.2 Random Walks

Trust aggregation approaches, however, are impractical in the case of a large OSN, where a user’s friends, friends-of-friends, etc., could quickly scale to a magnitude of thousands. For this reason, random walks have become a natural choice for researchers in the field of SRS [3, 8]. One of the first works on the subject is the *TrustWalker* system [1] which performs simple random walks on the trust graph, by defining transition and exit probabilities at each step of the walk. Neighbors, however, need not be chosen uniformly at random; in [2], the graph is initially traversed looking for the existence of strongly connected components. Then a nonuniform random walk is performed whose restarting probability depends on whether the currently active node is a member of a strongly connected component or not.

Random walks in the connected components of the graph assume the properties of *Markov Chains* (steady-state distribution, irreducibility, etc.). These properties have been further exploited by researchers as in [4], where a semi-supervised classification algorithm is applied in order to produce recommendations. The algorithm estimates the probability of a random walk starting at item y to terminate at the target user and these probabilities are considered to be markovian variables.

3 Design Aspects and Motivation

Although random walks in trust networks have been studied thoroughly, we believe there is still room for improvement. We must depart from the simple random walks that select their next step uniformly (or almost-uniformly) at random and introduce some bias toward “better” nodes. That is, we should discriminate our neighbors by increasing the transition probability toward similar users (defined in a recommendation context) and at the same time decreasing the transition probability toward dissimilar users.

3.1 Measuring Correlation

Unfortunately, trust and similarity are two concepts that do not necessarily coincide in SRS [10]. In the recommendation domain, two users are considered to be correlated

(similar) if they rate the same items in the “same” fashion. A number of metrics, derived from the statistical literature, measure how close two populations, i.e., U_x and U_y , are. U_x and U_y could be the ratings of user $u_x \in U$ and $u_y \in U$ on the same set of items I .

Statistical correlation has been extensively analyzed in the RS context and it has been found out that one of the most satisfactory metrics of correlation is the *Pearson Correlation Coefficient* [11], especially when the sets U_x and U_y coincide to a large extent. Unfortunately, this is not always the case in RS, particularly in sparse datasets. In such cases, other metrics like the *Log Likelihood* similarity or the *City-Block (Manhattan)* similarity yield better results.

3.2 Performing the Random Walk

Theoretically, better recommendations can be achieved if we walk toward more similar users (compared to selecting them uniformly at random). To further elaborate on this idea, we might consider the similarity metrics between a user and its direct neighbors in the trust network as samples of an unknown probability distribution that measures how close two neighbors actually are in their rating behavior. By moving toward like-minded neighbors (and not like-minded users as is the case with collaborative filtering), we increase our chances of getting a correct recommendation.

An obvious choice would be to pick the most similar neighbor each time. However, this is not the best strategy mainly because the ratings are not evenly distributed over all users and items in the dataset but follow a *Zipf Law* instead; a few users (items) issue a lot of ratings while most users (items) have only issued a few, belonging in the *long-tail* of the distribution. A deterministic algorithm would always pick the small slice of users and items with the most ratings and would consequently make recommendations from a restricted set of users, thus having a negative effect on the novelty and serendipity of the proposed items. Clearly, probabilistic algorithms allow for better exploration of the available choices contributing to the overall serendipity of the recommendation process.

The last issue that remains to be resolved is the fact that the target distribution we would like to sample from still remains unknown. For this reason, we first turn the similarity metrics into probabilities (by dividing each one with their sum) and then use an acceptance/rejection sampling algorithm to generate samples from.

4 The Biased Random Walk Algorithm

Our proposed random walk algorithm works in three phases. In the first phase and for each user, it retrieves from the user database all those users that have at least one rating in common with him/her (forming the set of the *Correlated Neighbors C*) and all those users that are trusted by the target user (forming the set of the *Trusted*

Neighbors T). Contrary to what might have been expected, these two sets are to a very large extent not overlapping. Therefore, a decision has to be made on which set of users to follow. A natural strategy would be to sample from each set based on its relative importance. That is, with a probability $P_T = \frac{|T|}{|C|+|T|}$ the next user in the walk is selected from T and with a probability of $P_C = 1 - P_T = \frac{|C|}{|C|+|T|}$ from C . In the first case, the next user is selected uniformly at random from T since the trust statements are binary. However, this rule does not hold for the second case, as users are correlated to one another to a different degree. It is this point where rejection sampling reveals its potential.

4.1 Rejection Sampling

The concept behind *Rejection Sampling* (or *acceptance-rejection* algorithm) is to use an easy-to-sample probability distribution $\mathcal{G}(x)$ as an instrument to sample from the unknown distribution $\mathcal{F}(x)$. $\mathcal{G}(x)$ is also referred to as the *proposal distribution*. Let $f(x)$, $g(x)$ be the respective probability distribution functions. The only prerequisite of this method is that the support of $g(x)$ dominates the support of $f(x)$ up to a proportionality constant c . That is, the following inequality must hold true:

$$f(x) \leq cg(x), c < \infty, \forall x \in \mathcal{X} \quad (3)$$

where \mathcal{X} denotes the sample space.

Next, a number n is drawn uniformly at random from $\mathcal{U}(0, 1)$ along with a sample $x_i \in \mathcal{X}$ according to $\mathcal{G}(x)$ ($x_i \sim \mathcal{G}(x)$). Then the inequality $n < \frac{f(x_i)}{cg(x_i)}$ is checked for its validity; if it holds, x_i is considered to be a valid sample drawn from $f(x)$, otherwise it is rejected and new samples n, x_i are drawn from the respective distributions.

Our recommendation algorithm performs a *Biased Random Walk* by applying the rejection sampling algorithm described earlier in order to decide its next step. For this reason, it is called *Biased RW-RS*. The target probability distribution $f(x)$ is constructed by dividing the similarity between the target user and each of its similar neighbors with the sum of their similarities. The uniform distribution $\mathcal{U}(x)$ is used as the proposal distribution and c is approximated by ensuring that the inequality $f(x) < cu(x)$ holds at each point. We then proceed to the rejection sampling method described in the function *RejectionSampling* (Fig. 1)

4.2 Terminating the Walk

An important decision to be made is when to stop the random walk. Stopping the walk early prevents the RS from exploring the user and item space, while stopping the

Require: Target User u_t

- 1: $u_c \leftarrow u_t$
- 2: **while** the walk is not terminated **do**
- 3: $(C, T) \leftarrow \text{SPLITNEIGHBORS}(u_c)$
- 4: Sample $n \sim \mathcal{U}(0, 1)$
- 5: **if** $n \leq \frac{|T|}{|C|+|T|}$ **then** ▷ u_c is the current node of the walk
- 6: Sample u_c uniformly at random from T
- 7: **else**
- 8: Sample $u_c \sim \text{REJECTIONSAMPLING}(T, u_c)$
- 9: **end if**
- 10: **end while**
- 11: **return** Visited nodes during the walk ▷ How often they were accessed
- 12:
- 13: **function** REJECTIONSAMPLING(S, u_c)
- 14: $sum \leftarrow 0$
- 15: **for** $u_n \in \text{Neighbors}(u_c)$ **do**
- 16: $s_{u_c, u_n} \leftarrow \text{SIMILARITY}(u_c, u_n)$
- 17: $sum \leftarrow sum + s_{u_c, u_n}$
- 18: **end for**
- 19: $\mathcal{G}(x) \leftarrow \mathcal{U}(\min s_{u_c, u_n}, \max s_{u_c, u_n})$ ▷ The proposal distribution becomes the uniform distribution defined over the space between the smallest and the largest similarity value
- 20: $c \leftarrow 0$
- 21: **for** $u_n \in \text{Neighbors}(u_c)$ **do** ▷ Turn Similarity into a Probability Distribution Function
- 22: $f(u_n) \leftarrow \frac{s_{u_c, u_n}}{sum}$
- 23: **if** $c < \frac{f(u_n)}{g(u_n)}$ **then**
- 24: $c \leftarrow \frac{f(u_n)}{g(u_n)}$
- 25: **end if**
- 26: **end for**
- 27: **repeat**
- 28: Sample $n \sim \mathcal{U}(0, 1)$
- 29: Sample $x_i \sim \mathcal{G}(x)$
- 30: **until** $n < \frac{f(x_i)}{c \cdot g(x_i)}$
- 31: **return** x_i
- 32: **end function**

Fig. 1 Biased Random Walk Rejection Sampling Algorithm

walk late has the risk of ending up in regions too far away from the target user. Since in most SRS the ratings' density is sparse and the global clustering coefficient of the social graph is very small, a simple probabilistic criterion is employed: with a fixed probability P_c (at each step), the walk continues and with probability $P_t = 1 - P_c$ the walk terminates (Bernoulli trial). The most widely adopted value for the termination probability is attributed to the PageRank algorithm [2] and is set at $P_t = 0.15$. After the walk termination, user nodes are ranked according to their relevance to the target user (how often they were visited during the walk) and recommendations are produced out of the most relevant ones.

5 Experiments

We have evaluated the performance of the *Biased RW-RS* algorithm into two different datasets. The first one was crawled from the Filmtrust website [12] and contains 33,526 ratings given by 1,919 users on 2,018 items, along with 1,591 trust statements. The second dataset was crawled from the Epinions website [13] and contains 664,824 ratings that 49,290 users have given to 139,738 items along with 487,182 trust statements. Both datasets are extremely sparse and the corresponding trust networks are extremely weak, following a Zipf law (Sect. 3). We have also examined three different correlation metrics as a similarity measure (line 16 of Algorithm 1); Pearson Correlation, Log Likelihood, and Manhattan Distance and we came to the conclusion that the last one is more suited for the datasets at hand.

5.1 Reference Systems

In order to better estimate the performance of the *Biased RW-RS* algorithm, we are presenting a number of reference RS (both traditional and social) and we are having them evaluated on the two datasets described above.

5.1.1 Baseline Systems

The purpose of the *Baseline Systems* is to estimate the relative improvements of the other systems. The *Random RS* would simply output a (uniformly) random value as a recommendation to each user, while the *Always-Max RS* would recommend each and every item to the target user with the biggest possible value.

5.1.2 Collaborative Filtering and Content-Based Approaches

The simple content-based and collaborative filtering recommendations are produced according to the widely adopted in the recommender systems' literature Resnick's formula [14]. The predicted rating that a particular target user would have given to a specific unseen item is determined by two factors: the target user's average rating on the other items he/she has evaluated so far and the ratings on the specific item given by the other users in the dataset:

$$\widehat{r_{u_t, i_t}} = \overline{r_{u_t}} + \frac{\sum_{i=1}^{|N|} w_{u_t, u_c} (\overline{r_{u_c}} - r_{u_c, i_t})}{\sum_{i=1}^{|N|} w_{u_t, u_c}} \quad (4)$$

where u_t is the target user and $u_c \in N$ all of his neighbors with whose the similarity value w_{u_t, u_c} can be computed.

5.1.3 Trust-Based Approaches

The *trust-based approaches* refer to the respective trust aggregation methodologies outlined in Sect. 2.1 (Eqs. 1 and 2). Especially for the MoleTrust case, the numerical suffix indicates the maximum propagation horizon.

6 Evaluation Metrics

6.1 Predictive Accuracy

Traditionally, the RS performance has been evaluated against the *Predictive Accuracy Metrics* [11]. Their purpose is to measure how close the predicted value $\widehat{r}_{u,i}$ is to a retained actual rating $r_{u,i}$. For this reason, the dataset is split into disjoint parts (sets) one of which is selected as the test set while the others form the training set. In our experiments, we have used the *tenfold cross-validation* model and the results on this category of metrics (Tables 2 and 3) are averaged for the 10 runs of the model.

The most widely adopted predictive accuracy metric is the *root mean square error* (RMSE), which is defined over a test set T as:

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{n=1}^{|T|} (\widehat{r}_{u,i} - r_{u,i})^2} \quad (5)$$

where $|T|$ is the cardinality of the test set.

A similar metric is the *mean absolute error* (MAE), which measures the difference between the output of the RS on a given input sample versus its expected value, averaged over all samples in T :

$$MAE = \frac{1}{|T|} \sum_{n=1}^{|T|} |\widehat{r}_{u,i} - r_{u,i}| \quad (6)$$

The two aforementioned metrics weight each prediction error the same and therefore favor users with more ratings. In order to introduce a trade-off between users with many ratings and cold-start users, Massa and Avesani [15] proposed the *mean absolute user error* which functions exactly like MAE; the only difference being that it first calculates the MAE over the ratings of a specific user and then computes the average of the MAE of all users:

Table 2 Predictive accuracy metrics: results on all users

Datasets	Filmtrust				Epinions			
Performance metrics	RMSE	MAE	MAUE	Coverage (%)	RMSE	MAE	MAUE	Coverage (%)
<i>A. Baseline</i>								
A.1 Random	1.53	1.25	1.26	100.00	1.94	1.61	1.63	100.00
A.2 Always-Max	1.35	1.00	0.90	100.00	1.57	1.01	0.97	100.00
<i>B. Collaborative filtering</i>								
(All neighbors)	0.88	0.70	0.68	93.65	1.07	0.81	0.82	79.57
<i>C. Content-based recommendation</i>								
(Nearest n items)	0.78	0.60	0.61	72.71	1.37	0.99	1.00	22.92
<i>D. Trust-based approaches</i>								
D.1 MoleTrust-1	0.97	0.73	0.74	18.64	1.23	0.91	0.95	25.58
D.2 MoleTrust-2	0.91	0.70	0.72	24.76	1.16	0.88	0.93	56.52
D.3 MoleTrust-3	0.89	0.69	0.70	27.14	1.12	0.85	0.89	70.89
D.4 TidalTrust	0.96	0.73	0.74	27.86	1.08	0.82	0.83	74.67
<i>E. Our recommender</i>								
E.1 Biased RW-RS	0.78	0.61	0.59	92.61	1.07	0.82	0.83	53.43

Table 3 Predictive accuracy metrics: results on cold-start users

Datasets	Filmtrust				Epinions			
Performance metrics	RMSE	MAE	MAUE	Coverage (%)	RMSE	MAE	MAUE	Coverage (%)
<i>A. Baseline</i>								
A.1 Random	1.51	1.22	1.22	100.00	2.00	1.67	1.67	100.00
A.2 Always-Max	0.80	0.49	0.51	100.00	1.56	0.94	0.93	100.00
<i>B. Collaborative filtering</i>								
(All neighbors)	0.80	0.64	0.63	82.98	1.09	0.82	0.82	69.46
<i>C. Content-based recommendation</i>								
(Nearest n items)	0.77	0.63	0.64	72.60	1.58	1.09	1.08	9.21
<i>D. Trust-based approaches</i>								
D.1 MoleTrust-1	1.46	1.20	1.02	10.94	1.49	1.09	1.09	7.49
D.2 MoleTrust-2	1.71	1.33	1.08	20.41	1.53	1.17	1.17	24.27
D.3 MoleTrust-3	1.22	0.87	1.33	24.56	1.06	0.82	1.08	76.25
D.4 TidalTrust	1.22	0.87	0.87	26.23	1.11	0.84	0.84	42.00
<i>E. Our recommender</i>								
E.1 Biased RW-RS	0.83	0.62	0.61	76.92	1.1	0.85	0.86	40.29

$$MAUE = \frac{\frac{1}{|M|} \sum_{u=1}^{|M|} |\widehat{r}_{u,i} - r_{u,i}|}{N} \quad (7)$$

where M are each distinct user's rating and N their overall number in T .

The *ratings' coverage* measures the percentage of ratings in the test set for which the system manages to make a prediction. It should be pointed out that an RS that exhibits satisfactory results in the statistical accuracy metrics is still considered to perform poorly if it manages to produce recommendations only for a handful of users or items. More formally, the *rating's coverage* is defined as

$$\text{Coverage} = 100 \frac{|T_R|}{|T|} \quad (8)$$

where $|T_R|$ is the cardinality of the set of the items for which the RS produced recommendations (generally, $T_R \subseteq T$).

Finally, it should be pointed out that the performance of the RS on the predictive accuracy metrics has been tested in two different user sets. The first one (Table 2) naturally includes all the users in the dataset while the second (Table 3) only includes the *cold-start users*; those users who have provided five ratings or less [15]. Cold-start users constitute a large portion of the user base of a RS and they could be viewed as the “newcomers” in the system. Definitely, the RS should be able to propose meaningful items from the very beginning in order to gain their confidence. Lastly, for the results presented in Tables 2 and 3, the standard deviation for RMSE, MAE, and MAUE was in the range of 0.01–0.03 for all systems, datasets, and views and in the 1–2% range for coverage.

6.2 Classification Accuracy

Classification accuracy metrics estimate the quality of the recommendations by measuring how frequently the RS makes good predictions [11]. This category of metrics is not evaluated on single withheld ratings but rather on a list of recommended items; for this reason the experimentation protocol has to be modified. Instead of splitting the whole dataset into disjoint sets, only the ratings of a particular target user are extracted and split into a training and a test set of a specific size (5 through 25 items in our experiments). Then, an equally sized list of items is presented to the target user and evaluated by the protocol. This process is repeated iteratively for all users and the results are averaged over all runs (Figs. 2 and 3). Since the recommendation list has to be of at least a certain size in order for the computations to be legitimate, this protocol cannot be run on the cold-start users alone. The results of the baseline systems are also not displayed because they have exhibited almost zero performance on this set of metrics.

Fig. 2 Classification accuracy metrics on the Filmtrust dataset.
a Precision. **b** Reach.
c Intra-list diversity. **d** Popularity-based item novelty

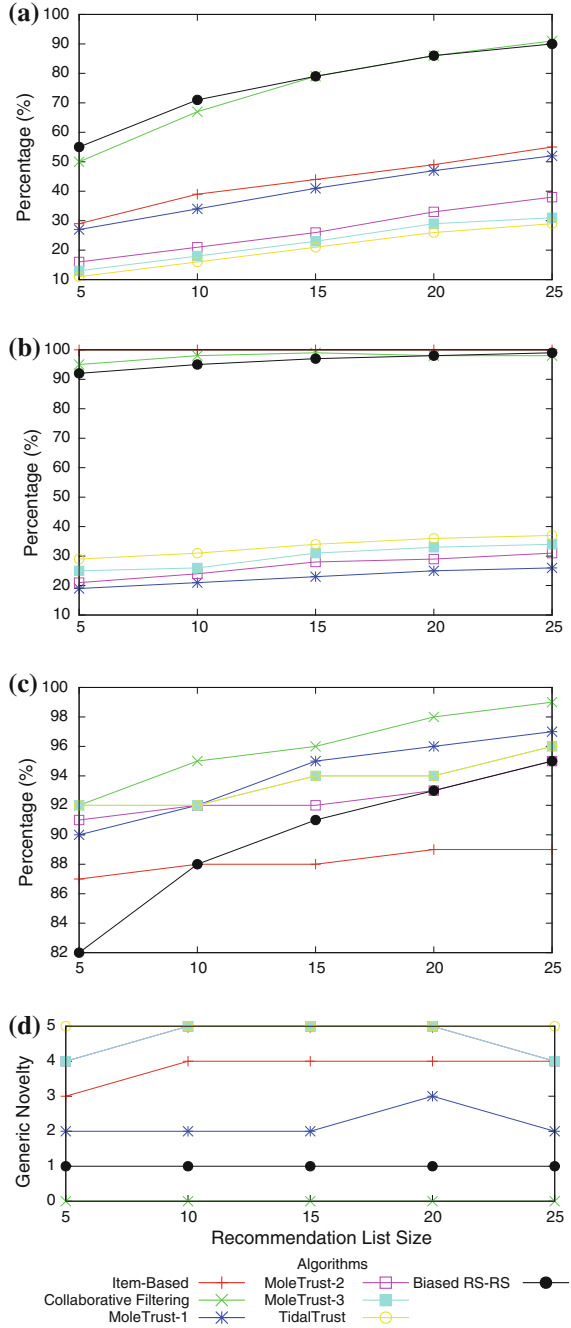
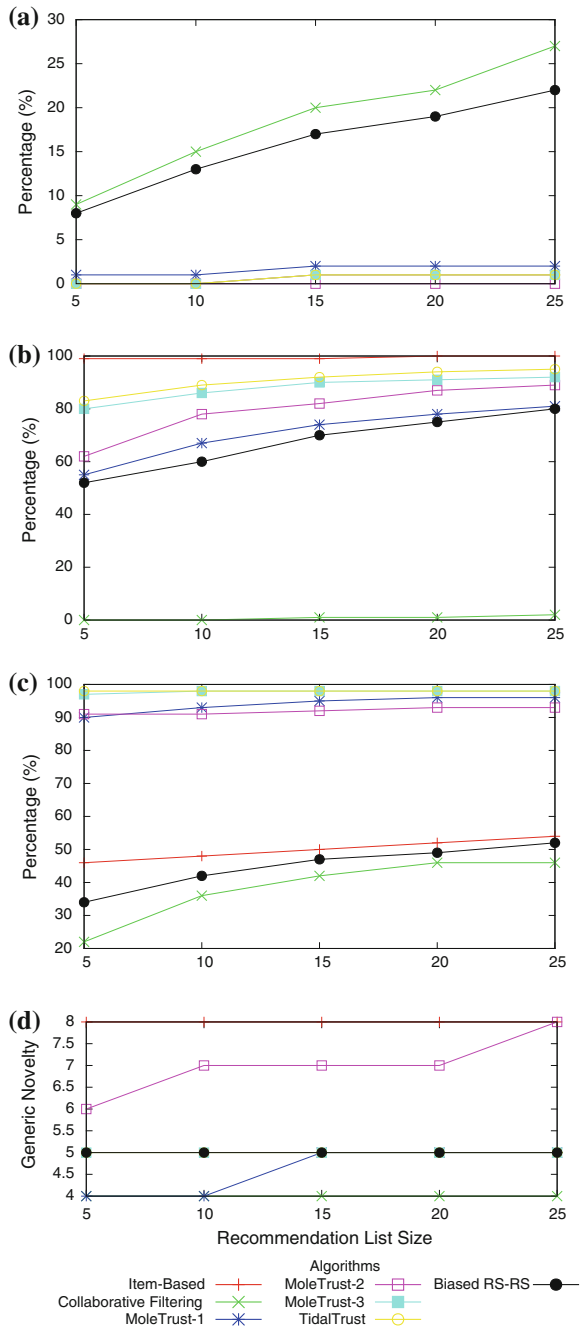


Fig. 3 Classification accuracy metrics on the Epinions dataset. **a** Precision. **b** Reach. **c** Intra-list diversity. **d** Popularity-based item novelty



The most common metric in this category is *Precision*. It measures the proportion of the relevant items selected in the list (N_{rs}) versus the total number of the selected items (N_s)

$$\text{Precision} = 100 \frac{N_{rs}}{N_s} \quad (9)$$

Alternatively, precision may be viewed as the probability that a selected item is relevant and it is most commonly expressed as a percentage.

A metric similar to the ratings' coverage discussed in Sect. 6.1 is *Reach*, or the percentage of the users for whom the RS manages to produce recommendations. Again, a recommender system that exhibits a high precision in its proposals is still considered to perform poorly if it manages to do so only for a handful of users. More formally, *Reach* is defined as

$$\text{Reach} = 100 \frac{|U_R|}{|U|} \quad (10)$$

where $|U_R|$ is the cardinality of the set of the users for which the RS produced recommendations and $|U|$ is the cardinality of the set of users in the system (generally, $U_R \subseteq U$).

6.3 Novelty and Diversity

This set of metrics tries to quantify the obviousness and the ordinariness of the recommendations a particular user receives. Generally, commonplace recommendations are considered to be of low quality even if they are correct in terms of both the prediction and the classification accuracy [11]. These two metrics are evaluated on a list of recommended items and for this reason the experimentation protocol of the previous subsection is applied and the results are displayed on the same set of Figs. 2 and 3

Novelty measures the extent to which an item (or a set of items) is new when compared with those items that have already been consumed by a user (or a community of users). Several models of item novelty have been proposed in the literature; in our experiments, we have used *the generic popularity-based item novelty* [16], which is defined as

$$\text{novelty}(i) = I(i) = -\log_2 p(i) \quad (11)$$

where $p(i)$ is the probability of observing item $i \in I$ in the result set. In our case, we considered this probability to be analogous to the number of ratings this item has received ($|R_i|$) proportional to the total number of ratings in the dataset ($|R|$)

$$p(i) \sim \frac{|R_i|}{|R|} \quad (12)$$

Diversity, on the other hand, measures how different the items of a recommendation list are from one another. A list of items that are relevant but very similar to each other is considered to be very ordinary and thus of low quality. In our experiments, we have used the *Intra-list Diversity* metric defined as

$$\text{diversity}(L) = \frac{2}{|L|(|L| - 1)} \sum_{k < n} d(i_n, i_k) \quad (13)$$

where L is the list of the recommended items, $i_n, i_k \in L$ and $d(i, j)$ is an item distance measure. As we have been using the Manhattan distance measure which takes values in the $[0, 1]$ interval, the results of the intra-list diversity are normalized on the percentage scale (Figs. 2 and 3).

7 Results

A first observation is that the *Biased RW-RS* algorithm is comparable to the collaborative filtering approach in all of the predictive accuracy metrics on the whole users view, despite being a social method. In general, Social RS exhibit poor behavior on coverage and this is attributed to the fact that the trust network in both datasets is very sparse; as a result, their exploration ability is greatly impacted. However, the *Biased RW-RS* manages to overcome this difficulty by probabilistically deciding at each step to either pick a trust neighbor or a similar user. Therefore, it is far superior in terms of coverage in the Filmtrust dataset and on the average of the SRS in the Epinions dataset.

Furthermore, for the cold-start users, the *Biased RW-RS* algorithm is among the most efficient approaches in the Filmtrust dataset for this special case of users, clearly outperforming the other social methods, while the performance of the other RS (traditional and social) deteriorates evidently. Again, in the Epinions dataset, our algorithm manages to keep a steady performance in terms of the MAE and RMSE metrics, while at the same time offering an adequate coverage on the ratings of the test set.

The *Biased RW-RS* algorithm is the most efficient social method at the precision metric on both datasets. Although collaborative filtering seems to be slightly better in the Epinions dataset, it performs very poorly on the reach metric (around 1–2% on all list sizes), meaning that it is able to produce accurate recommendations only for a tiny slice of the users. Trust approaches, on the other hand, are able to produce recommendations for more users; however, these predictions are far from accurate (precision is less than 5% for all trust metrics on all list sizes on the Epinions dataset and less than 40% on the Filmtrust dataset) because user correlation is not taken into account. Another notable observation for the trust approaches is that their Reach on the Filmtrust dataset is about one-third compared to the Epinions dataset, even though the trust network of the former dataset is denser than the latter (global clustering coefficient characteristic of Table 1). This phenomenon is attributed to

the fact that only 38 % of the users in the Filmtrust dataset participate in the trust network, while the same figure for the Epinions dataset is 68 %. As a conclusion, even the smallest user engagement in the trust network is sufficient for the SRS to make recommendations.

The trust approaches also demonstrate the best results in terms of both the novelty of the recommendations and the diversity of the items in the recommendation list. However, this behavior should not be studied independently from Precision; diverse and novel predictions are of no use if they are not relevant to the user. On the other hand, correlation-based approaches (item-based and collaborative filtering RS) make recommendations of items that are very obvious and to a large extent very similar to one another. For this reason, these systems exhibit very poor novelty, which is also illustrated in the respective Figs. 2d and 3d.

In all, the results indicate that our system achieves recommendation accuracy similar to the traditional collaborative approaches while showing better novelty and diversity, due to the incorporation of social aspects in its recommendation mechanism.

8 Conclusion

In this chapter, we have presented a novel approach toward SRS, a random walk recommender system based on rejection sampling. Our contribution is an algorithm, *Biased RW-RS*, which is based on a novel idea in neighborhood selection; it deviates from the standard view of all trust neighbors as equally probable and models their similarity to the target user as a probability distribution. Since this probability distribution is unknown for each user, it is approximated by using readily applied tools from the statistical literature and more specifically of the rejection sampling algorithm. Generally, the results on the reference datasets are encouraging and in accordance to our claims.

We are also taking into consideration the fact that our system does not exhibit a steady performance lead in the Epinions dataset. We attribute this behavior to the peculiarities of this specific dataset; its greater sparsity and the fact that it is not domain specific (as opposed to the Filmtrust dataset). Therefore, our algorithm should be further adapted in the direction of addressing the aforementioned observation.

References

1. Jamali M, Ester M (2009) TrustWalker: a random walk model for combining trust-based and item-based recommendation. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD'09, New York, NY, USA, ACM, pp 397–406
2. Abbassi Z, Mirrokni VS (2007) A recommender system based on local random walks and spectral methods. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop

- on Web mining and social network analysis. WebKDD/SNA-KDD'07, New York, NY, USA, ACM, pp 102–108
3. Yuan Q, Chen L, Zhao S (2012) Augmenting collaborative recommenders by fusing social relationships: membership and friendship. In: Recommender systems for the social web, vol 32 of Intelligent Systems Reference Library, Springer, Berlin, pp 159–175
 4. Zhang Y, Wu Jq, Zhuang Yt (2009) Random walk models for top-n recommendation task. *J Zhejiang Univ Sci A* 10:927–936
 5. Singh AP, Gunawardana A, Meek C, Surendran AC (2007) Recommendations using absorbing random walks. In: North East Student Colloquium on Artificial Intelligence (NESCAI)
 6. Golbeck JA (2005) Computing and applying trust in web-based social networks. Ph.D. thesis, College Park, MD, USA, AAI3178583
 7. Andersen R, Borgs C, Chayes J, Feige U, Flaxman A, Kalai A, Mirokni V, Tennenholtz M (2008) Trust-based recommendation systems: an axiomatic approach. In: Proceedings of the 17th international conference on world wide web. WWW'08, New York, NY, USA, ACM, pp 199–208
 8. Konstas I, Stathopoulos V, Jose JM (2009) On social networks and collaborative recommendation. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval. SIGIR'09, New York, NY, USA, ACM, pp 195–202
 9. Alexandridis G, Siolas G, Stafylopatis A (2013) A biased random walk recommender based on rejection sampling. In: Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM 2013), Niagara Falls, Canada
 10. Massa P, Avesani P (2007) Trust metrics in recommender systems. *Int J Semant Web Inf Syst*
 11. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. *ACM Trans Inf Syst* 22(1):53
 12. Golbeck J, Hendler J (2006) FilmTrust: movie recommendations using trust in web-based social networks. In: Consumer communications and networking conference, 2006. CCNC 2006, 3rd IEEE, vol 1, pp 282–286
 13. Massa P, Bhattacharjee B (2004) Using trust in recommender systems: an experimental analysis. In: Proceedings of iTrust2004 international conference, pp 221–235
 14. Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithm for collaborative filtering. In: Proceedings of the 14th conference on uncertainty in artificial intelligence, pp 43–52
 15. Massa P, Avesani P (2009) Trust metrics in recommender systems. In: Golbeck J (ed) Computing with social trust, Human Computer interaction series. Springer, London, pp 259–285
 16. Castells P, Vargas S, Wang J (April 2011) Novelty and diversity metrics for recommender systems: choice, discovery and relevance. In: International workshop on diversity in document retrieval (DDR 2011) at the 33rd European conference on information retrieval (ECIR 2011)