Comparing the Predictive Capability of Social and Interest Affinity for Recommendations

Alexandra Olteanu[1], Anne-Marie Kermarrec[2], and Karl Aberer[1]

[1] Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland
[2] INRIA Rennes-Bretagne Atlantique, Rennes, France

Abstract. The advent of online social networks created new prediction opportunities for recommender systems: instead of relying on past rating history through the use of collaborative filtering (CF), they can leverage the social relations among users as a predictor of user tastes similarity. Alas, little effort has been put into understanding when and why (e.g., for which users and what items) the *social affinity* (i.e., how well connected users are in the social network) is a better predictor of user preferences than the *interest affinity* among them as algorithmically determined by CF, and how to better evaluate recommendations depending on, for instance, what type of users a recommendation application targets. This overlook is explained in part by the lack of a systematic collection of datasets including both the explicit social network among users and the collaborative annotated items. In this paper, we conduct an extensive empirical analysis on six real-world publicly available datasets, which dissects the impact of user and item attributes, such as the density of social ties or item rating patterns, on the performance of recommendation strategies relying on either the social ties or past rating similarity. Our findings represent practical guidelines that can assist in future deployments and mixing schemes.

Keywords: Social affinity, Interest affinity, Recommender systems, Collaborative Filtering, Evaluation

1 Introduction

Recommender systems are inescapable in a wide range of web applications, e.g. Amazon or Netflix, to provide users with books or movies that match their interest. Accurate recommendations generate returns of investments up to 30% due to increased sales [24]. Many such systems rely on collaborative filtering (CF) approaches that recommend items based on user rating history. Concomitantly, the rising popularity of social networks has provided new opportunities to filter out relevant content for users. For instance, recommendation services like Epinions, Last.fm or BeerAdvocate are enhanced with virtual social networks.

As a result, existing works have proposed both pure social recommenders (SR)[3] that only leverage the social ties among users [33], and hybrid approaches

[3] For readability, social refers to both trust and social.

that either augment the CF recommendation engine with social guidelines [39, 31] or incorporate CF mechanisms into a social recommendation engine [20].

A common practice in evaluating such approaches is to resort to *(i)* one [42, 45, 35, 22, 20, 25, 30], sometimes two [39, 31] datasets and *(ii)* global averages for the metrics of choice. Alas, this has made it difficult to draw generalizable conclusions on the effectiveness of leveraging the social ties for recommendations compared with CF across datasets of different nature.

Furthermore, the use of global metrics[4] to evaluate and compare the recommendation approaches may be inconclusive as they provide little insight into *when* and *why* the approaches succeed or fail [13]. Although the impact of the parameters of a recommendation strategy has been often inspected [9, 39, 22, 41, 29, 20, 7], little systematic effort has been devoted into understanding how various user or item attributes are affecting the performance [2], and none of such analyses, to our knowledge, have included SR approaches.

Orthogonal to designing better hybrid approaches that combine SR and CF features, our goal is to gain insight into the relative benefits of each of these approaches that, in turn, can guide future deployments and mixing schemes. To do so, we perform an extensive empirical analysis that dissects the recommendation performance, measured by precision and coverage, and does a fine grained comparison across various user and item classes on six *publicly available* datasets including both the ratings information and the social network among users (§3). All datasets are medium to large-scale and exhibit various properties regarding user social ties and items ratings. We focus on the two ends of the problem spectrum, which places on the one side the *interest affinity* among users (resp. items), as algorithmically determined by CF from user rating history, and at the other side the *social affinity* as inferred from users social network by *pure* SR (§2). Our analysis addresses two main questions:

(1) Are global metrics able to reflect the performance of a given recommendation strategy across various settings? Our analysis shows that one cannot rely on global metrics to assess a given recommender performance not only across all datasets but also within each dataset, across different classes of users or items. Even a slight change in the global average might hide important changes in the performance distribution across a dataset demographics. One may thus need to understand and optimize the performance on a specific demographic subset depending on the application specifics (e.g., for a beer recommendation service, it might be more important to be accurate in the recommendations to experienced and, thus, harder to please users [34]).

(2) Are there user or item attributes that hint at the CF (interest affinity) performance with respect to SR (social affinity)? In our results, we find that when the basis of formulating connections among users stems from *plain* friendship, rather than from sharing interests, SR leads to less precise recommendations. Further, items likeability (the rating they received on average) and user selectiveness are good predictors of the recommendation performance: relying on *social affinity* leads to more precise predictions for highly liked items, while for

[4] Metrics that are computed or averaged for all predictions.

indulgent users (that typically give high ratings) leveraging the *interest affinity* for items similarity is best. More results are discussed in (§3).

2 Problem Definition

Typically, a recommender task is to predict ratings for unseen items to users. To do so, a set of items I, a set of users U, and a set of items $I_u \subseteq I$ rated by each user u with a rating $r_{u,i}$ on a Likert scale from 1 to 5 is considered. If the recommender system exploits the social ties among users, for each user u a set of friends F_u is assumed. This paper looks at the predictive capability of social ties (SR) compared to the one of items or users rating similarity (CF) for items recommendation.

2.1 Comparison Framework

We conduct our study using a comparison framework that implements a recommendation template under which, to make a recommendation for user u on target item i, two main steps are performed[5]: (1) identify the set of similar users (resp. items) with u (resp. i) and (2) compute weighted aggregates of their ratings on i (resp. from u) according to the similarity with u (resp.i). On top of it, we implement the main building blocks of SR and CF as used for comparison in literature [3, 20, 21, 35, 25, 23]. Specifically, we implement (a) item- and user-based CF variants as often used as reference point by previous work [20, 35, 38, 25], and (b) a SR approach that aggregates the ratings similarly with CF, yet, instead of deriving users affinity based on how similar they rated items in the past, it does so based on their social ties. Next we describe each approach and motivate our choices.

Collaborative Filtering (CF) approaches are usually grouped in two main classes: *neighborhood-* and *model-based* [12]. Model-based variants have received lot of attention as their accuracy was considered superior, yet neighborhood-based CF, though simpler, remains competitive [11]. Further, they exploit different patterns in data, none of them consistently out-perform the other: model-based CF is typically effective at estimating the overall model related to all items simultaneously, while neighborhood-based CF better captures local associations in data [6]. This trait makes neighborhood-based CF suitable for our purpose to compare the predictive capability of *interest affinity* (inferred based on implicit similarity links as determined by CF) and *social affinity* (computed based on explicit social links among users). Further, neighborhood-based CF offers a simple and intuitive template for recommendation to easily implement a pure SR-based approach on top of it and fairly compare the two under the same setting.

We use common variants of the two main types of neighborhood-based CF: user- and item-based CF. Briefly, for each user u (resp. item i) a neighborhood UN_u (resp. IN_i) of users (items) similar with u (resp. i) is built and their ratings on the target item i (resp. from active user u) are aggregated as:

$$p_{u,i} = \frac{\sum_{v \in UN_u} sim(u,v) r_{v,i}}{\sum_{v \in UN_u} sim(u,v)} \tag{1}$$

[5] As in neighborhood-based CF [18]

for user-based CF, where $sim(u, v)$ is the similarity between users u and v, as estimated by the Pearson correlation of the ratings given by u and v on the same items[6]; respective, $p_{u,i} = \frac{\sum_{j \in IN_i} sim(i,j) r_{u,j}}{\sum_{j \in IN_i} sim(i,j)}$ for item-based CF, where $s(i, j)$ is the Pearson correlation of the ratings received by i and j from the same users.

Social Recommendation (SR) In contrast to CF[7], when the ratings received by target item i are aggregated according to Eq. (1), SR weights them based on the social affinity between the active user (i.e., the user for which we want to make a prediction) and the users that have rated item i in the past.

Social Affinity (relatedness) of two nodes in a social graph can be estimated using random walks (RWs) [28], which have been used for both friend [5, 26] and item recommendations [43, 20, 14]. In short, for each prediction, we run RWs on the social graph that start at user u needing a recommendation on item i, and stops when they either reach a user v that have rated the target item i, or have performed a maximum number of steps k_{max}[8]. We denote a RW stopping condition with $s_{v,i,k}$, which is *true* if $i \in I_v$ or $k >= k_{max}$, meaning that the RW stops at v. Then, the social affinity between u and user v that rated the target item i is the probability to reach v using different paths and number of steps: $P(X_{u,i} = v) = \frac{\sum_k P(X_{u,i,k}=v)}{\sum_{w \in U} \sum_k P(X_{u,i,k}=w)}$, where the random variable $X_{u,i}$ represents the nodes that rated item i and can be reached at any step of the RW starting at node u, while $X_{u,i,k}$ represents only the subset of nodes reachable at step k:

$$P(X_{u,i,k} = v) = \sum_{w \in U} P(X_{u,i,k-1} = w) P(X_w = v) \tag{2}$$

where $P(X_{u,i,0}) = 1$ and X_w the random variable to pick a friend of node w. For unweighted graphs (as those used in our evaluation), we have: $P(X_w = v) = \frac{1}{|F_w|}$.

Thus, the probability to step on node $v \in F_w$ at step $k+1$ after being at node w at step k is $P(X_{u,i,k+1} = v | X_{u,i,k} = w, \overline{s}_{w,i,k}) = P(X_w = v)$, where $X_{u,i,k}$ is the random variable for nodes that can be reached at step k when looking for i, $\overline{s}_{w,i,k}$ is the negation of $s_{w,i,k}$, and $P(X_{u_i,k+1} = v | X_{u,i,k} = u, s_{w,i,k}) = 0$ to complete the probability distribution. To also complete the specification of the probability distribution in Eq. (2), we define a final state $\perp$, to which the RW goes when it terminates: $P(X_{u,i,k} = \perp) = 1 - \sum_{v \in U} P(X_{u,i,k} = v)$.

To determine if we performed enough RWs to make an admissible prediction, after each RW we compute the variance $\sigma^2 = \frac{\sum_{j=1..T} (r_j - \overline{r})}{T}$ in the results of all the walks [20], where T is the number of successful walks[9], r_j is the result returned by the j-th RW, and $\overline{r}$ is the mean of the results return by the RWs. If the variance σ^2 converges to a constant (i.e., the variance after $j+1$ walks varies with less than $\epsilon = 0.0001$ from the variance after j walks), or the total number of (successful and unsuccessful) walks reaches the maximum number of walks $T_{max} = 1000$,

[6] Note that we also consider only positive correlations [20]

[7] For brevity, when referring to both user-based and item-based CF, we use only CF.

[8] Set to 6 based on the "six-degree of separation" assumption [36] that most of the nodes are reachable within 6 hopes [20]

[9] A random walk is successful if it encounters a user that have rated the target item.

Dataset	Users	Items	Ratings	Social Links	Links Type
Ciao	12,375	99,762	284,086	237,350	direct
Epinions1	49,290	139,738	664,824	487,181	direct
Epinions2	22,166	296,277	922,267	355,813	direct
Epinions	132,000	755,760	13,668,319	841,372	direct
Flixster	786,936	48,794	8,196,077	7,058,819	symmetric
Douban	129,490	58,541	16,830,839	1,692,952	symmetric

Table 1. Datasets Figures

we stop from running more RWs. Then, to make a prediction, in Eq. (1), we replace the similarity between active user u and user v which have rated item i with their relatedness in the social network: $p_{u,i} = \sum_{\{v \in U | i \in R_v\}} P(X_{u,i} = v) r_{v,i}$.

3 Empirical Analysis

In this section we perform an extensive analysis that juxtaposes the SR (social affinity) and CF (interest affinity) as predictors for item recommendation, structured in three parts. First, we present a comprehensive characterization of the datasets. Second, we apply global metrics to evaluate the recommendation strategies, and examine if they capture the performance variation across various settings. Finally, we do a fine grained analysis of the impact of user and item properties on the performance, organized as a set of questions about CF and SR properties. These questions are largely inspired by admitted properties of CF or SR, such as, CF performs better on users for which it has more information [15, 4, 8], the recommendation accuracy decreases towards the long-tail items (i.e., less popular items) [40], or SRs are superior on *cold start* users [20, 33].

3.1 Metrics and Experimental Setup

To evaluate the recommendation performance, we use the well-known *leave one out* strategy. Specifically, we remove from the dataset only the rating we want to predict and leave the other ratings and social network unchanged. Then, we compare CF and SR along two popular metrics: (1) The *coverage* measures a recommendation strategy ability to make predictions, and it is the number of ratings the system succeeded to make divided by the total number of ratings that it tried to predict. (2) The *Root Mean Square Error (RMSE)* captures the average error between the predictions and the real ratings, measuring the recommendation precision: $RMSE = \sqrt{\frac{1}{N}\sum(r_{u,i} - p_{u,i})^2}$, where N is the number of predictions, $r_{u,i}$ the real rating given by u to item i, while $p_{u,i}$ is the prediction. Note that the smaller the RMSE is, the more precise the recommendations are.

Albeit RMSE ability to gauge the performance for pervasive top-k recommendations is debated [10], it best fits our purpose to measure performance shifts across classes of items/users. The accuracy metrics deemed suitable to evaluate top-k performance, are biased towards the performance on preferred items (i.e., high ratings) [19]. Moreover, many recommender systems that leverage the social ties optimize for RMSE [44], making our analysis convenient to compare with.

Two approaches are used to report RMSE and coverage values for a set of users/items: (1) compute the RMSE (resp. coverage) over all the predictions to users (or for items) in the set; or (2) compute the RMSE (resp. coverage) for each

Dataset	Ratings Per User	Ratings Per Item	Avg. Degree	Mean Rating	Median Rating
Ciao	22.9	*2.8*	**19.1**	4.16	4
Epinions1	13.4	4.7	9.8	3.99	4
Epinions2	41.6	3.1	16	3.97	4
Epinions	103.5	18.0	*6.3*	**4.67**	**5**
Flixster	*10.4*	167.9	8.9	*3.8*	4
Douban	**129.9**	**287.5**	13.0	3.84	4

Table 2. Dataset Statistics. Bold marks the highest value per column, while italic the lowest.

item/user separately and average the results over all users (resp. items) in the set. While the first measures the overall performance on estimating the ratings, the second weights each user (resp. item) equally measuring how good the predictions are on average for each user (resp. item) in the set. We measured both, yet, due to space limitations, when the two variants lead to similar conclusions we show only results with the second one; both are included otherwise. Finally, when measuring how a certain user (resp. item) property impacts the results, we group the users (resp. items) by logarithmically binning them regarding the property value, and then compute the performance for each bin[10].

3.2 Datasets Characterization

We conduct our analysis on 6 real world publicly available datasets including both ratings and a social network (figures are summarized in Table 1):

Epinions is a popular product review site where people rate products and build lists of trusted users whose reviews they find useful. We use two rating datasets from Epinions: one is collected by the authors of [32] around 2006 (noted *epinions1*), and one is collected in May 2011 by the authors of [42] (noted *epinions2*). In addition to product ratings, in Epinions, users can also rate product reviews. We also use a dataset, made available by Epinions.com to the authors of [32] containing ratings on product reviews, instead of ratings on products (noted *epinions*). In all datasets the ratings are on a scale from 1 to 5.

Douban is a Chinese product review site that represents one of the largest online communities in China. As in Epinions, users rate and review products in order to receive recommendations. In addition, at the date of crawling, it provided a Facebook-like social networking service [31].

Ciao defines itself as a *multi-million-strong online community* in which users critically review and rate millions of products. It provides the same functionality as Epinions (i.e., users can both rate products and indicate the trusted users)[41].

Flixster is a large social movie rating service that allows users to create Facebook-like friendship relations and share ratings [22], which are from 0.5 to 5 (with a step of 0.5). To ensure uniformity across the analyzed datasets, we round the ratings to the next integer so as to obtain ratings on a 1 to 5 scale.

[10] We use logarithmic binning (in base 4) to account for the fact that some values in the degree, popularity, or activity distributions are frequent while others are not. A linear binning leads to bins with few or no points.

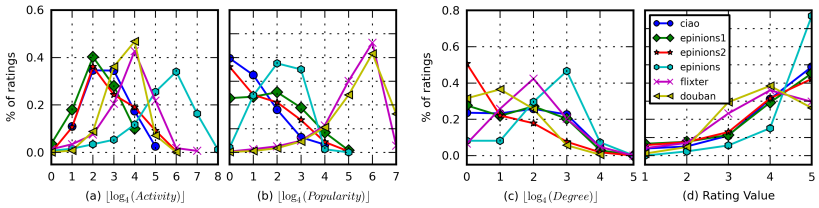


Fig. 1. Distribution of ratings as function of: (a) user activity; (b) item popularity; (c) user degree; (d) rating value

Data Statistics. We want to understand the properties of the datasets we analyze, the resemblance among them, as they might explain the performance variations across them. Table 2 highlights basic statistics for each dataset.

Rating Distributions. Fig. 1 shows the rating distributions across user and item properties, and the rating value. In Fig. 1(a) we notice similar patterns across datasets with only little variation (for larger datasets, the level of user activity at which the peak number of ratings is produced is shifted towards higher ranges). In contrast, the rating distribution according to item popularity, Fig. 1(b), varies greatly: while in some datasets (*ciao*, *epinions1*, *epinions2*) the highest fraction of ratings is given to unpopular items, in others (the largest ones) this is accounted for popular items. Fig. 1(c) also shows that while in *flixster* and *epinions* most ratings are given by moderately social connected users, in other datasets a higher number of ratings is credited to lower degree users. Looking at rating distributions according to the rating value, Fig. 1(d), we see that in all datasets the values are skewed towards higher ranges (peaking around 5).

Item Distributions. We observe similar patterns across all datasets: Fig. 2(a) illustrates that with only one exception (*epinions*) the cold start items (with only few ratings) represent a significant fraction of all items. Fig. 2(b) shows that in all datasets most of the items received on average a rating of 3 or 4.

User Distributions. SR is believed to address *cold start* users, as it does not require them to rate items for making predictions, but only to be connected in the social network. Given that in some datasets the number of *cold start* users is

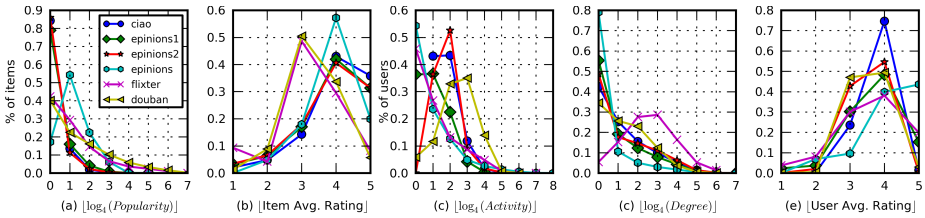


Fig. 2. The distribution of items as a function of (a) item popularity and (b) average rating per item, and the distribution of users as a function of (c) user activity, (d) user (out-)degree and (e) average rating per user.

Dataset	User CF	Item CF	Social
Ciao	**1.144** (0.410)	1.285 (0.318)	1.252 (**0.626**)
Epinions1	**1.186** (0.512)	1.428 (0.463)	1.362 (**0.663**)
Epinions2	**1.164** (**0.483**)	1.361 (0.395)	1.406 (0.365)
Epinions	**0.466** (0.930)	0.602 (0.579)	0.559 (**0.951**)
Flixster	1.013 (0.969)	**0.889** (**0.991**)	1.349 (0.985)
Douban	**0.784** (0.996)	0.809 (**0.997**)	1.037 (0.894)

Table 3. Overall performance. In each cell we report RMSE (Coverage) computed over all the ratings in the dataset. Bold highlights the best value on each row.

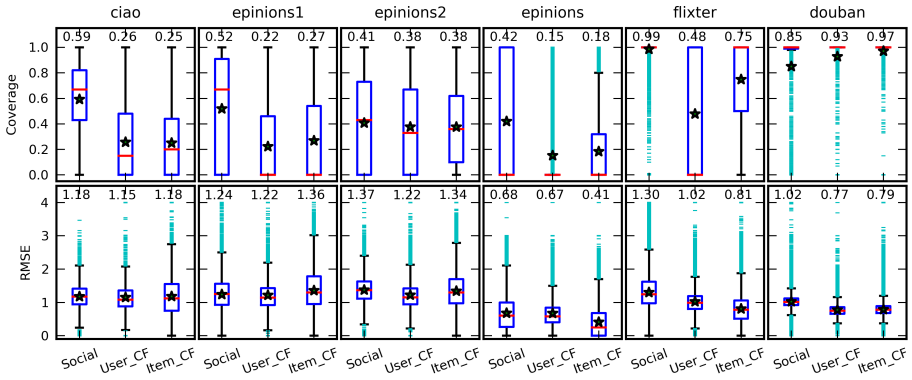
significant (roughly 50% [20]), improving on this set of users might significantly impact the overall performance. Thus, on average such approaches were found to outperform CF [20, 33]. Yet, when the percentage of cold start users is not significant, this might not be the case. Fig. 2(c) shows that while in some datasets (*epinions*, *flixster*) cold start users are a significant percentage, this is clearly not the case in others (*douban*, *ciao*). Additionally, regardless of their fraction, cold start users always produce a minor fraction of ratings (see Fig. 1). In Fig. 2(d), we notice that, except *flixster*, the number of low degree users is larger than the number of cold start users, which in turn might affect SR overall performance. Finally, Fig. 2(e) shows that, on average, users tend to give higher rating values.

Correlations. We also check the correlation among item and user properties (item popularity, user activity and degree, and the average rating received by an item or given by a user). As in general we found low or no correlation, we report only on statistically significant ($p < 0.01$) moderate Pearson correlations ($|r| \geq 0.2$). We found moderate and positive correlations among users degree and their level of activity in *ciao* ($r = 0.59$), *epinions1* ($r = 0.45$) and *epinions*($r = 0.36$). In *flixster* ($r = 0.43$), *douban* ($r = 0.35$) and *epinions* ($r = 0.30$) there is a positive correlation between items popularity and the ratings they got, i.e, popular items tend to obtain higher ratings. Item popularity also correlates negatively with users level of activity in *flixster* and *douban* ($r = -0.20$ in both datasets), i.e., active users are more inclined to rate unpopular items. While in *douban* there is a negative correlation ($r = -0.29$) between users level of activity and the ratings they give on average, indicating that active users are more likely to give lower ratings; in *epinions* popular items tend to get higher ratings ($r = 0.31$).

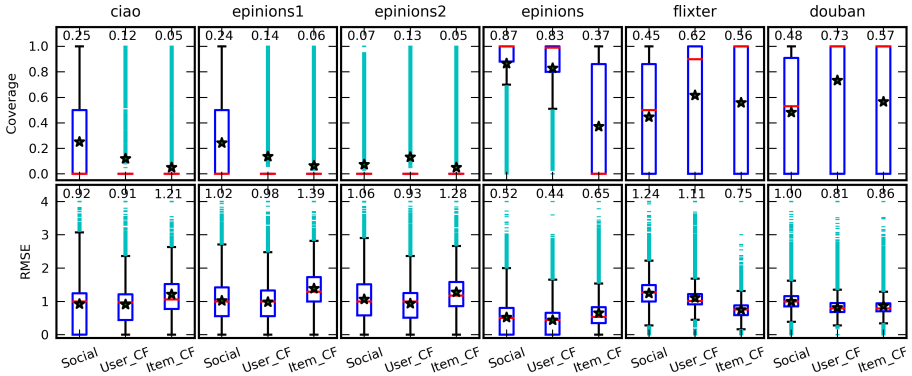
We will see in the next sections how these varying data properties explain the different performance numbers obtained when aggregating the results differently (e.g., user-oriented vs. item-oriented evaluation) within and across datasets.

3.3 Overall Performance Characterization

A common practice in recommender systems evaluation is to show how their performance varies with approach-dependent parameters. Yet, even when there are correlations between the parameter values and performance level, it is difficult to know, for instance, if the improvements hold for the entire population, or only for some subgroups. Thus, we want to observe if there is a trivial relationship between the experimental results obtained through globally computed metrics that summarize the performance, typically used to evaluate recommendation systems



(a) Per-user distributions



(b) Per-item distributions

Fig. 3. Results Distribution: The boxplots divide the data, except outliers (the blue lines), in four equal buckets. A data point displays the performance on a particular user (resp. item). The redline splitting the boxplot is the median, while the star is the average performance (also plotted above each boxplot).

[20, 22, 35, 39], and the averaged performance at user (resp. item) level. Table 3 reports the globally computed metrics (*rating-oriented evaluation*) per dataset and approach. For error rates, with only one exception (i.e., *flixster*), user-based CF performs best across all the datasets. In terms of coverage, there is no clear winner: SR performs best for *ciao*, *epinions1* and *epinions*, while user-based CF for *douban* and *epinions2*, and item-based CF for *flixster*. Next, we check if these results are also confirmed by the *user (resp. item)-oriented evaluations* (§3.1) which measures how well an approach does on average per user (resp. item). In Fig. 3 the boxplots show the shape of the average performance distribution for users (resp. items), its central value, and variability.

User-oriented evaluation. Fig. 3(a) shows the per-user performance variation across datasets. Though it mostly confirms the overall results (in terms of winners) for most datasets, there are exceptions in which SR, resp. item-CF, fares

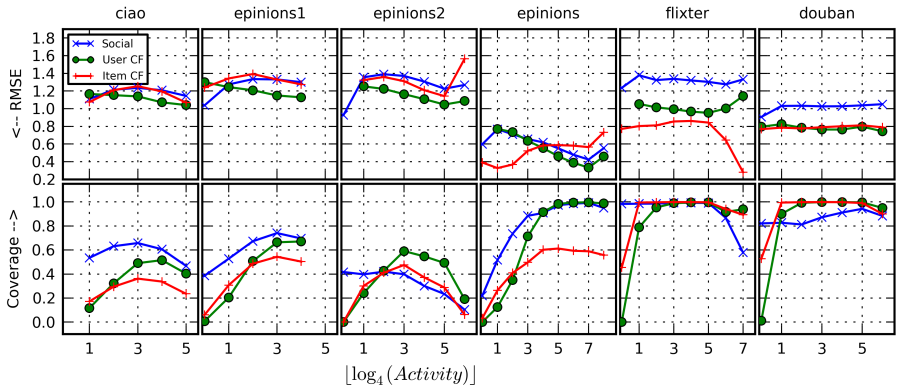


Fig. 4. Performance as a function of user activity: (top) average RMSE per user; (bottom) average coverage per user.

better than the globally computed metrics indicate in Table 3: e.g., the coverage on *flixster*, where the fraction of unsocial users is lower than that of cold start users, and RMSE on *epinions*, where there is a higher fraction of items with similar ratings, than of users giving similar ratings.

Item-oriented evaluation. Similarly, barring the coverage on *flixster* and *douban*, Fig. 3(b) also confirms (in terms of winners) the figures in Table 3. Yet, we notice that except *epinions* and *flixster*, in all the other datasets both the distributions and the average coverage values are significantly shifted towards lower ranges regarding the user-oriented evaluation, which is explained in part by the much higher fraction of unpopular items than of cold start users that these datasets exhibit.

This demonstrates that it is difficult to rely on global metrics to assess or explain a given recommender performance, a finer granularity has to be applied; and that indeed no general conclusion can be drawn regarding the relative superiority of a given recommendation method over another, not only across datasets but also within each dataset.

3.4 In-Depth Performance Characterization

We aim to understand the benefit of each approach under a variety of settings. In this regard, we address a set of questions about the properties of CF and SR, some of which are well embedded in the conventional wisdom:

Does CF fare better for users (resp. items) with more ratings? The belief is that CF does better when a user has rated more items [15, 8]. To test it, we analyze how CF performs as users are more active (have rated more items). Fig. 4 shows that users' level of activity impacts the ability to make predictions (the coverage) similarly across all approaches: being more active helps only until some threshold after which rating more items either does not help (*epinions*, *douban*) or can even be harmful (*epinions2*). Further, while rating more items tends to help user-based CF to make precise prediction (in *epinions* and *flixster*

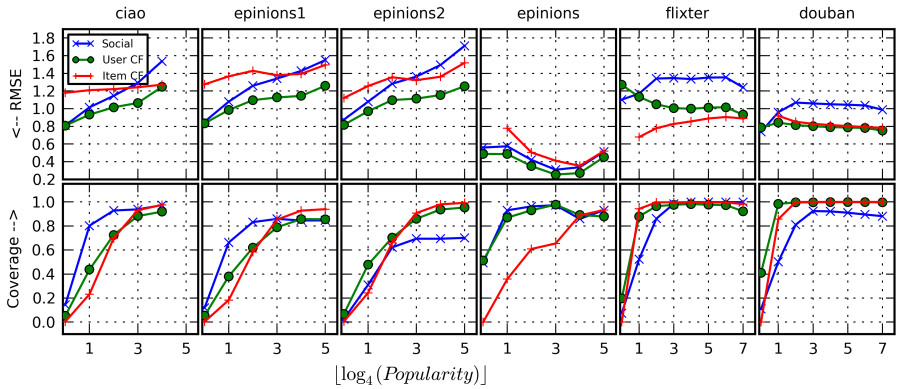


Fig. 5. Performance as a function of item popularity: (top) average RMSE per item; (bottom) average coverage per item.

after slightly improving for a while, the error increases again), item-based CF has a more inconsistent pattern. Looking at the relative performance of CF regarding SR (barring cold start users, i.e., the first bin on the $\log_4$ scale), we notice that users level of activity impacts user-based CF and SR similarly in terms of both coverage and RMSE. Exceptions are the coverage results on the datasets that exhibit no correlation among users social degree and their level of activity (*douban*, *flixter*).

As with more ratings per user, the belief is that more ratings per item help CF [40]. To challenge it, we look how CF performs with the number of ratings per item. Fig. 5 shows that the average coverage per item is improving as items are more popular only until some threshold when they plateau. In contrast, for *ciao*, *epinions1*, *epinions2* (datasets with a small number of ratings per item, Table 2) the predictions are less precise as the items are more popular, *invalidating* the belief. Checking the relative performance of CF regarding SR, we notice that more ratings per item helps CF to increase its precision regarding SR. The only exception is *epinions* (to easily spot the patterns, follow on y-axis the distance between points corresponding to the same bin but with distinct approaches).

Does SR fare better for cold star users? The belief is that SR deals better with cold start users [20] (with less than 5 items rated [16]) as it only requires them to be connected to other users to make predictions. Indeed, Fig. 4 shows that SR achieves better coverage for these users (leftmost bins) across datasets. Yet, this is not always the case when it comes to precision (RMSE). For instance, we observe that for *flixster* and *douban* (when the social ties stem from friendship), CF attains a better precision for all users, including cold start ones.

Does SR fare better for users with more social connections? Intuitively, more social information available should help SR. To check this, we study how SR performs across users with various social degrees. Fig. 6 shows that higher degrees help improve the coverage only until users are moderately connected (have at least 5 connections), after which linking to more users seems to bring little or no

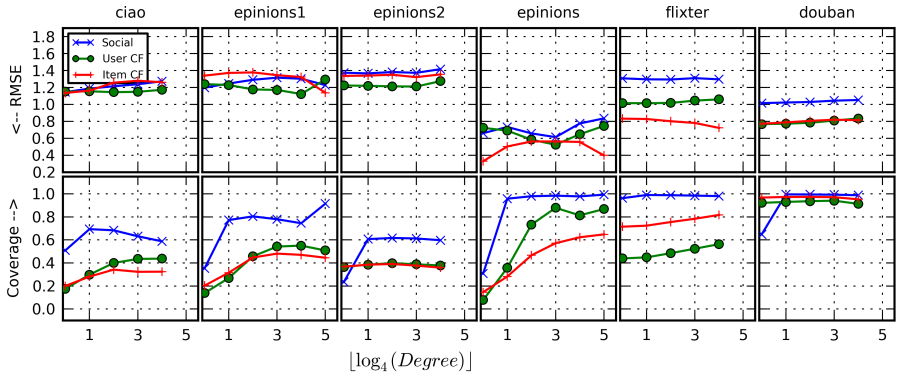


Fig. 6. Performance as a function of node degree: (top) average RMSE per user; (bottom) average coverage per user.

benefit for SR, even declining on *ciao*. Neither SR's precision improves as users are more socially active: it either slightly decreases, or plateau. This means that having too many friends might also introduce noise. This hints that many social ties might not reflect as much friendship, similarity or trust. However, on most datasets higher degrees tend to have a weak to no impact on SR's precision. Further, as with the level of activity, baring the low degree users, the social degree impacts user-based CF and SR in a similar way, in particular for those datasets in which the degree correlates with the level of activity.

Is CF doing better on low degree nodes? Since CF does not leverage the social links to make predictions, it should not be affected by their absence, and, thus, should perform better on *unsocial* (low-degree) users. Yet, Fig. 6 shows that CF succeeds to obtain a better coverage on unsocial users only for *douban* and *epinions2*. For RMSE, while on some datasets CF does better on *unsocial* users, when there is a correlation between user degrees and how many items they rate (*ciao*, *epinions1*, and *epinions*), it performs comparable with SR.

Is SR's Precision re. CF Smaller on Facebook-like Networks? The process of creating connections primarily based on "plain" friendship (Facebook-like) does not necessarily correlate with one's opinions as it is orthogonal to a product recommendation task. Yet, when the basis of forming connections is to connect with people whose opinions one shares, there might be more agreement in how users rate the same items. Indeed, this distinction is clearly visible in our results (Fig. 6 to 8): while SR fares comparable with CF in terms of RMSE in Epinions datasets and *ciao*, in Facebook-like *flixster* and *douban* CF significantly outperforms SR. In addition, being more socially active has little to no impact on the results obtained for *flixster* and *douban* (Fig. 6). Thus, this indicates that the underlying nature of the network and whether or not the connections are related or orthogonal to the recommendation task is an important factor as well.

Is the performance independent of users selectiveness or items likeability? Only few studies hint at the relation between user selectiveness [34]

or items likeability and recommendation performance. Yet, in Fig. 7 we notice consistent patterns across datasets, in particular, for RMSE. In all datasets item-based CF is less precise when items are either liked (received high ratings), or disliked (received low ratings), while SR and user-based CF are less precise for users that are either very selective (giving mostly low ratings) or indulgent (offering mostly high ratings). Also note how similarly both the user and item average rating impacts the precision across all datasets (i.e., leading to similar curves for all datasets). This is surprising as it indicates that the users (resp. items) average rating is predictive for the recommendation approach precision. It is also worth noting that user-based CF and SR precision (although with slightly different values) follow almost identical curves. Yet, as Fig. 7 illustrates, for coverage the patterns are not consistent across all datasets.

4 Related Work

Collaborative Filtering (CF) has been widely used by major commercial applications such as Amazon, Movielens, or Netflix [24, 27, 1]. These methods leverage users rating history and predict the rating of a target item and a source user by looking at the ratings on the target item given by similar users, *user-based approaches* [17], or at what ratings items similar to the target one have received from the source user, *item-based approaches* [38]. Yet, relying solely on CF is ineffective when dealing with large numbers of items, given the sparsity of the user-item ratings matrix. *Cold start* users and items are particularly affected, CF often failing to make predictions in such cases (i.e., leading to a low *coverage*).

Social recommender systems(SR) In contrast, SR systems leverage users social ties to make predictions [46, 33, 37, 16, 35], assuming that these reflect common tastes or interests. SR systems deal better with *cold start* users, as they require users only to be connected to other users in the social network, and do not have to wait for users to grow a rating history to make predictions. Alas, while these systems tend to achieve better coverage, they can also suffer due to sparse ratings and sparse trust relations. Thus, in order to consider the ratings of users that are not directly connected, various approaches propagate the trust among their users [46, 16, 33, 35]. Yet, in these cases the recommender might end up considering ratings of weakly trusted users, thus affecting the precision [20].

Social-enhanced collaborative approaches incorporate social factors to the collaborative framework by tailoring the rating similarity based on the social ties [25]; making predictions based on friend ratings weighted by the level of trust, and integrating them in the CF framework [29]; adding social regularization factors to matrix factorization recommendation techniques by constraining a user inferred taste (her feature vector) with the average taste of her friends, and the similarity with each of them [31], thus making her feature vector depend on those of her friends [22], or by accounting for the social ties heterogeneity [39].

In contrast, *collaborative-enhanced social approaches* implement a social-based framework that falls back on CF when trusted users did not rate the target item. TrustWalker enhances a social-based approach with item-based CF [20], and employs a random walk model that first tries to exploit the social network by looking for the ratings on the target item at trusted nodes (*trust-based ap-*

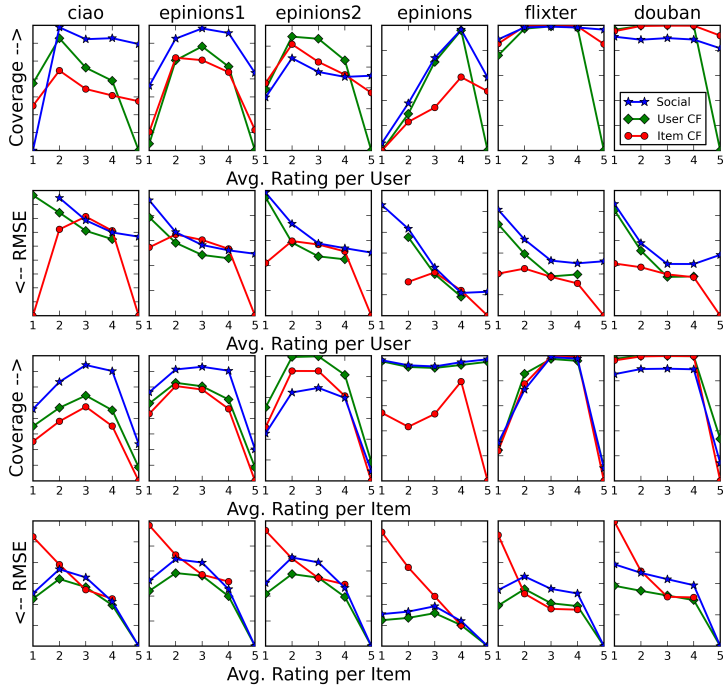


Fig. 7. Performance as a function of average rating value per item and per user.

proach). Yet, as the random walk advances, if a rating on this item is not found, the likelihood to return the rating of a similar item (*item-based approach*) increases. TrustWalker acts in extreme settings as a pure SR approach when the random walk never stops for similar items, and as pure item-based CF when the walk never starts (navigating the same problem spectrum with us).

5 Concluding Remarks

We conducted an in-depth empirical analysis on six *publicly available* datasets to study the respective merits of the *interest affinity*, as derived by CF, and the *social affinity*, reflecting how well connected users are in the social graph, for items recommendations. We focused on the building blocks of the analyzed strategies, without aiming to exhaustively inspect all possible implementations, as we argue that their understanding can better guide more complex deployments. Our study conveys that the level of user activity, item popularity or the density and nature of the underlying social network are as many characteristics that can impact the performance of recommendation systems. One needs to understand the dataset demographics and optimize the performance based on each application specificities. We make a case for hybrid approaches, that dynamically adapt as the system evolves and the properties of user and item change over time.

Acknowledgements

We thank Stefan Bucur, Yelena Mejova and Saket Sathe for their valuable feedback on earlier versions of this work. This work was partially supported by the grant Reconcile: Robust Online Credibility Evaluation of Web Content from Switzerland through the Swiss Contribution to the enlarged European Union.

References

1. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 2005.
2. G. Adomavicius and J. Zhang. Impact of data characteristics on recommender systems performance. *ACM Trans. Manage. Inf. Syst.*, 2012.
3. G. Adomavicius and J. Zhang. Stability of recommendation algorithms. *ACM Trans. Inf. Syst.*, 2012.
4. X. Amatriain. Mining large streams of user data for personalized recommendations. *SIGKDD Explor. Newsl.*, 2013.
5. L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, 2011.
6. R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. *SIGKDD Explor. Newsl.*, 2007.
7. A. Bellogín, I. Cantador, F. Díez, P. Castells, and E. Chavarriaga. An empirical comparison of social, collaborative filtering, and hybrid recommenders. *ACM Trans. on Intel. Sys. and Tech.*, 2013.
8. R. Burke. Integrating knowledge-based and collaborative-filtering recommender systems. In *Workshop on AI and Electronic Commerce*, 1999.
9. W. Chen, W. Hsu, and M. L. Lee. Making recommendations from multiple domains. In *KDD*, 2013.
10. P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*, 2010.
11. L. M. de Campos, J. M. Fernandez-Luna, J. F. Huete, and M. A. Rueda-Morales. Measuring predictive capability in collaborative filtering. In *RecSys*, 2009.
12. C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*. Springer, 2011.
13. M. Ekstrand and J. Riedl. When recommenders fail: predicting recommender failure for algorithm selection and combination. In *RecSys*, 2012.
14. F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. on Knowl. and Data Eng.*, 2007.
15. N. Golbandi, Y. Koren, and R. Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *WSDM*, 2011.
16. J. Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, University of Maryland, 2005.
17. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 1992.
18. J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*, 1999.
19. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 2004.

20. M. Jamali and M. Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *KDD*, 2009.
21. M. Jamali and M. Ester. Using a trust network to improve top-n recommendation. In *RecSys*, 2009.
22. M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, 2010.
23. A.-M. Kermarrec, V. Leroy, A. Moin, and C. Thraves. Application of random walks to decentralized recommender systems. *Principles of Distributed Systems*, 2010.
24. J. Konstan and J. Riedl. Recommended for you. *Spectrum, IEEE*, 2012.
25. I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR*, 2009.
26. D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 2007.
27. G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 2003.
28. L. Lovász. Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty*, 2(1):1–46, 1993.
29. H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR*, 2009.
30. H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, 2008.
31. H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, 2011.
32. P. Massa and P. Avesani. Trust-aware bootstrapping of recommender systems. In *ECAI Workshop on Recommender Systems*, 2006.
33. P. Massa and P. Avesani. Trust-aware recommender systems. In *RecSys*, 2007.
34. J. J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *WWW*, 2013.
35. S. Meyffret, L. Médini, and F. Laforest. Trust-based local and social recommendation. In *RSWeb*, 2012.
36. S. Milgram. The small world problem. *Psychology today*, 1967.
37. G. Pitsilis and S. J. Knapskog. Social trust as a solution to address sparsity-inherent problems of recommender systems. In *Recommender Systems and the Social Web*, 2009.
38. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, 2001.
39. Y. Shen and R. Jin. Learning personal + social latent factor model for social recommendation. In *KDD*, 2012.
40. H. Steck. Item popularity and recommendation accuracy. In *RecSys*, 2011.
41. J. Tang, H. Gao, and H. Liu. mtrust: Discerning multi-faceted trust in a connected world. In *WSDM*, 2012.
42. J. Tang, H. Liu, H. Gao, and A. Das Sarmas. etrust: understanding trust evolution in an online world. In *KDD*, 2012.
43. S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *WWW*, 2011.
44. X. Yang, H. Steck, Y. Guo, and Y. Liu. On top-k recommendation using social networks. In *RecSys*, 2012.
45. X. Yang, H. Steck, and Y. Liu. Circle-based recommendation in online social networks. In *KDD*, 2012.
46. C.-N. Ziegler. *Towards Decentralized Recommender Systems*. PhD thesis, Albert-Ludwigs-Universitat Freiburg, 2005.