An Empirical Comparison of Social, Collaborative Filtering, and Hybrid Recommenders

ALEJANDRO BELLOGÍN, IVÁN CANTADOR, FERNANDO DÍEZ, PABLO CASTELLS AND ENRIQUE CHAVARRIAGA
Universidad Autónoma de Madrid

In the Social Web, a number of diverse recommendation approaches have been proposed to exploit the user generated contents available in the Web, such as rating, tagging, and social networking information. In general, these approaches naturally require the availability of a wide amount of the above user preferences. This may represent an important limitation for real applications, and may be somewhat unnoticed in studies focusing on overall precision, in which a failure to produce recommendations gets blurred when averaging the obtained results or, even worse, is just not accounted for, as users with no recommendations are typically excluded from the performance calculations. In this paper, we propose a coverage metric that uncovers and compensates for the incompleteness of performance evaluations based only on precision. We use this metric together with precision metrics in an empirical comparison of several social, collaborative filtering, and hybrid recommenders. The obtained results show that a better balance between precision and coverage can be achieved by combining social-based filtering (high accuracy, low coverage) and collaborative filtering (low accuracy, high coverage) recommendation techniques. We thus explore several hybrid recommendation approaches to balance this tradeoff. In particular, we compare, on the one hand, techniques integrating collaborative and social information into a single model, and on the other, linear combinations of recommenders. For the last approach, we also propose a novel strategy to dynamically adjust the weight of each recommender on a user-basis, utilizing graph measures as indicators of the target user's connectedness and relevance in a social network.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering, retrieval models*

General Terms: Algorithms, Experimentation, Performance.

Additional Key Words and Phrases: Recommender Systems, collaborative filtering, hybrid recommenders, social networks, graph theory, random walk, user coverage.

1. INTRODUCTION

1.1 Motivation

Information systems where users share different types of feedback on products and services are commonplace nowadays. Users' input creates rich explicit and implicit relations among items based on several forms of interaction: reviews, comments, ratings, and other types of feedback in the system, such as click-through data and browsing time. The richness and availability of user input and interaction data at worldwide scale has further boosted more research and development of complex technologies that analyze and

Authors' addresses: Alejandro Bellogín, Iván Cantador, Fernando Díez, Pablo Castells, Enrique Chavarriaga; Universidad Autónoma de Madrid, Calle Francisco Tomás y Valiente 11, 28049 Madrid, Spain; e-mail: {alejandro.bellogin, ivan.cantador, fernando.diez, pablo.castells}@uam.es, jes.chavarriaga@estudiante.uam.es.
Permission to make digital/hard copy of part of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date of appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.
© 2011 ACM 1073-0516/01/0300-0034 $5.00

exploit this data in elaborate ways, opening new scenarios for the creation of novel services and business models. Personalized recommendation technologies are a characteristic example of this trend. Recommender Systems (RS) aim to automatically find the most useful products or services for a particular user, providing a personalized list of items for that user based on different inputs and attributes of users and items [ADOMAVICIOUS & TUZHILIN, 2005]. The state of the art recommender system technologies build on implicit relationships between users and items derived from different data sources, such as the content of the items or the preferences of similar-minded users. They exploit these types of relationships to a significant degree of effectiveness, playing a consistent role in successful businesses such as Amazon[1] or Netflix[2]. Seeking new horizons beyond current recommender technologies, the research community is investigating how to exploit other attributes and contexts that are currently reaching critical mass in terms of scale, density, and availability in such systems. An example of what has rapidly attracted considerable interest from the research community and industry in recent years is the "social" context, as a consequence of the outburst of social systems in the so called Social Web or Web 2.0 [KONSTAS et al., 2009; MA et al., 2008; LIU & LEE, 2010].

Social recommenders are proving to be a good alternative and/or complement to Collaborative Filtering (CF) methods. CF algorithms, one of the most widespread recommendation approaches so far [KOREN & BELL, 2011], make use of user preferences for items to find similar-minded people (or similar items) according to the observed user interaction with the items, and present suggestions on this basis. Social recommenders, on the other hand, exploit the social context, represented as links between users in a community, to suggest interesting items. Although social recommenders perform very well in some situations, they naturally require the availability of a social context. This requirement introduces an important limitation in real applications, namely, the inability to produce recommendations for users, when insufficient social links are known to the system. This limitation may go somewhat unnoticed in studies focusing on overall recommendation accuracy, in which a failure to produce recommendations gets blurred when averaging the obtained results or, even worse, is just not accounted for, as users with no recommendations are typically excluded from the performance calculations.

[1] Amazon, Electronic commerce company, http://www.amazon.com
[2] Netflix, Movie and television rental service, http://www.netflix.com

Besides the accuracy of delivered recommendations, the ratio of users for which a recommender is able to deliver a recommendation, which we shall call user coverage, is indeed a dimension one may also want to pay attention to. It is often the case that accuracy and coverage involve a tradeoff –e.g., one may enhance accuracy by withholding difficult recommendations [HERLOCKER et al., 2004]. In this work we suggest the use of a performance metric based on the tradeoff between precision and user coverage when building and evaluating recommender systems. We explore the combination of social and collaborative algorithms into hybrid recommendation approaches as a means to achieve a balanced tradeoff between the two. We investigate different techniques for this combination, ranging from algorithms that integrate the two information sources into a single model (e.g., graph-based recommenders) to linear combinations of recommenders (e.g., weighted hybrid recommenders). Furthermore, for the latter case, we explore a novel approach to dynamically adapt the weight of each recommender on a per-user basis.

We conduct the empirical part of our study in two different experimental setups: one in which only users with friends –i.e., those users with an explicit (direct) social relationship– are considered in the test set (social evaluation), and another in which users with and without friends are tested (collaborative-social evaluation). The experimental study is conducted using one of the datasets provided for the CAMRa Challenge [SAID et al., 2010]. These datasets were gathered from the Filmtipset and Moviepilot communities, and contain social links between users, movie ratings, movie reviews, review ratings, comments about actors and movies, movie directors and writers, lists of favorite movies, moods, and links between similar movies. Despite the proliferation of online information systems supporting both preference data and friendship links, the CAMRa data is, to the best of our knowledge, the only publicly available dataset where this variety of user data is provided, meeting our experimental requirements.

1.2 Contributions

In this work, we make the following contributions:

- We implement an array of state-of-the-art recommenders and propose **modifications and adaptations of recommendation approaches** based on those algorithms: personalized random walk with restart, popularity based on friends, and combinations of social and collaborative filtering recommenders. We will show that some of these algorithms deliver competitive performance under different evaluation conditions. In order to better understand the relation between

accuracy and the amount of social context on a per-user basis, we compare the accuracy of social recommenders against collaborative filtering systems, and analyze two metrics, named **user coverage** and **hitrate**. The former considers how many users can receive a recommendation, and the latter, how many users receive a relevant recommendation. Besides, we conduct two different evaluations – including and excluding users with an empty social context– so that we can compare different recommenders' behavior for those metrics.

- When building a social recommender system, we have to take into account a **tradeoff between precision and coverage**. We address this problem, first, with recommenders that exploit a single source of information, and second with hybrid recommenders, in order to analyze whether hybridization obtains better balanced results than single, standard recommenders. We have found that social-based recommenders obtain good performance results, at the expense of low user coverage. On the other end of the spectrum is the performance (and coverage) of user-based recommenders, since their coverage is higher, but the performance is lower. We have found that hybrid algorithms improve the performance of collaborative methods and the coverage of social recommenders.
- Weighted hybrid recommenders have two important shortcomings. First, the optimal weight has to be found empirically by relying on the recommender performance, dataset characteristics, etc, which are subject to change. Second, the weight should not be the same for all users in the system. We propose a novel approach for **dynamic hybrid recommendation** that makes use of indicators based on graph measures in order to adaptively weight the combined recommenders for certain users. In this situation, by dynamically selecting the weight for each method on a per-user basis, the results outperform those of simple hybrid recommenders.

1.3 Structure of the paper

The rest of the paper is structured as follows. Section 2 describes the evaluated recommenders. We present some well-known state-of-the-art collaborative filtering, social and hybrid recommender systems, and present extensions and adaptations of some of these recommenders in order to exploit social information. Section 3 discusses the results obtained when the user coverage metric was introduced in order to perform a fair comparison among the different techniques. Before that, we present some evaluation considerations taken into account in the experiments conducted, such as the evaluation

methodology used, and the adaptation of some recommenders for their use with the Filmtipset dataset. Section 4 describes related works that exploit different social characteristics applied to recommendation, and investigates new evaluation metrics in recommender systems. Finally, section 5 summarizes the paper with some conclusions and potential future research lines.

2. EVALUATED RECOMMENDERS

In this paper, we investigate different types of recommender systems: collaborative filtering, social-based, and hybrid approaches. We present in this section an overview of these recommenders, including both state-of-the-art algorithms, and extensions and adaptations of some of them in order to exploit social information.

2.1 Notation

In the rest of the paper we shall use the following notation. We denote by $\mathcal{U} = \{u_1, \dots, u_{\mathrm{L}}\}$ the set of users, and by $\mathcal{I} = \{i_1, \dots, i_{\mathrm{M}}\}$ the set of items; we reserve the letters u, v for users, and i, j for items. $B[u,k]$ denotes the set (with size k) of neighbors of u, $\mathcal{I}_m \subseteq \mathcal{I}$ represents the subset of items rated by user u_{m}. The function $pos(u,i)$ gives the position of the item i in the recommended list for user u. We represent $rat(u,i)$ as the rating given by user u to item i; if the rating is unknown the symbol ϕ is used. We use $\overline{rat}(u,\cdot)$ to represent the average rating given by a user u; similarly, $\overline{rat}(\cdot,i)$ is the average rating received by item i. The predicted score of a user for an item is represented as $s(u,i)$. Similarity functions are denoted as $sim(a,b)$, where a and b may be a pair of users or items.

2.2 Collaborative Filtering Recommenders

Collaborative Filtering is often based on explicit ratings, in such a way that systems record reviews/votes/evaluations that users give to certain items, thus, creating a user-item matrix in which each user is associated with a row, and each item, with a column. The value in each cell of that matrix is the rating of a particular user-item pair if it is available, or a value not used in the rating scale otherwise; for instance, 0 is typically used when the rating scale is 1-5. In implicit CF systems, on the other hand, the available information corresponds to records of user's activities and effective usage of items (e.g., browsing, purchases and downloads of products). Recommendation algorithms devised for explicit ratings can be applied to such implicit data by transforming non weighted user-item pairs (where pairs are assigned, for instance, an access frequency value, or a list of access timestamps) into (single valued) user-item-vote tuples. This processing may

simply consist of recording binary votes (presence or absence of activity), or more refined techniques, which take advantage of further available information, such as item access frequency, duration, type of access (e.g., inspect vs. buy), etc [LINDEN et al., 2003; CELMA, 2008; BALTRUNAS & AMATRIAIN, 2009]. In this paper, we focus on user- and item-based CF exploiting explicit ratings. More specifically, we include nearest-neighbor and matrix factorization algorithms in our study.

User-based nearest-neighbor CF techniques compare the target user's preferences with those of other users to identify a group of "similar-minded" people (usually called *neighbors*). Once this group has been identified, items highly rated by this group are recommended to the target user. More specifically, the predicted score $s(u_m, i_n)$ for user u_m and item i_n is estimated as follows:

$$s(u_m, i_n) = \mathrm{C} \sum_{v \in B[u_m,k]} sim(u_m, v) \times rat(v, i_n) \quad (1)$$

where C is a normalization factor. Similarity between users can be calculated by using different metrics: Pearson and Spearman's correlations, cosine-based distance, and others [ADOMAVICIOUS & TUZHILIN, 2005]. In this work, we use Pearson's correlation, which is defined as:

$$sim(u, v) = \frac{\sum_i \big(rat(u,i) - \overline{rat}(u,\cdot)\big)\big(rat(v,i) - \overline{rat}(v,\cdot)\big)}{\sqrt{\sum_i \big(rat(u,i) - \overline{rat}(u,\cdot)\big)^2} \sqrt{\sum_i \big(rat(v,i) - \overline{rat}(v,\cdot)\big)^2}} \quad (2)$$

Additionally, item-based CF techniques recognize patterns of similarity between the items themselves, instead of between user choices like in user-based strategies. Item-based CF looks at each item in the target user's list of rated items, and finds other items that seem to be "similar" to that item. The item similarity is usually defined in terms of correlations of ratings between users [ADOMAVICIOUS & TUZHILIN, 2005]. More formally, the score $s(u_m, i_n)$ predicted by this method is estimated as follows:

$$s(u_m, i_n) = \mathrm{C} \sum_{j \in I_m} sim(i_n, j) \times rat(u_m, j) \quad (3)$$

As stated in [SARWAR et al., 2001], adjusted cosine similarity has proven to obtain better performance than other item similarities, such as cosine distance, and Pearson's correlation, when they are used in equation 3. Adjusted cosine similarity subtracts the user's average rating from each co-rated pair in the standard cosine formulation:

$$sim(i,j) = \frac{\sum_u \big(rat(u,i) - \overline{rat}(u,\cdot)\big)\big(rat(u,j) - \overline{rat}(u,\cdot)\big)}{\sqrt{\sum_u \big(rat(u,i) - \overline{rat}(u,\cdot)\big)^2}\sqrt{\sum_u \big(rat(u,j) - \overline{rat}(u,\cdot)\big)^2}} \quad (4)$$

[BARMAN & DABEER, 2010] proposes a recommender where the items suggested to a user are the most popular ones among her set of similar users. The authors use a binary matrix model as input data (known rating values as 1's and unknown ones to 0's), and pick those items having the maximum number of 1's among the active user's top k neighbors –where neighbors are selected according to some similarity measure (e.g., Pearson's correlation). In this paper, we consider this method as a baseline, and we extend it by considering also non-binary data, that is, we rank the items for each user according to the following quantity:

$$\gamma(u_m, i) = |\{v \in B[u_m, k]: rat(v, i) \neq \phi\}| \quad (5)$$

In this way, each item is ranked according to how popular it is among the set of user's neighbors. Once a ranking has been generated using the quantities calculated in (5), we compute a score by transforming the item position with the following equation:

$$s_N(u_m, i_n) = 1 - \frac{pos(u_m, i_n)}{N} \quad (6)$$

This transformation implies that the first item in the ranking receives a score very close to one, while the last item ($pos(u_m, i_n) = N$) gets a score of 0. If the item does not appear in the ranking generated for that user, a score of 0 is assigned. In this formulation, we assume that the length of the list is N, and, thus, we may trim the returned list at some level N, or assume N to be exactly the length of the generated recommendation list . Note that these scores are not in the range of ratings, but lie in the interval [0, 1), and because of that, they do not have a clear interpretation for rating value prediction, but they can be used for generating item rankings. It would not be difficult to modify the range of this transformation, using, for instance, techniques from rank aggregation [FERNÁNDEZ, VALLET, CASTELLS, 2006]. However, since in our experiments we are interested in measuring ranking precision, this function could be considered a valid transformation.

Additionally, matrix factorization techniques have been successfully applied to the problem of rating prediction. These methods factorize the user-item ratings matrix into a product of two matrices, in order to identify latent semantic factors [KOREN & BELL, 2011]. These methods map the preference data (user's ratings for items) to a latent factor space of a particular dimensionality. The specific value of the dimension of the output space is chosen a priori and has impact on the final performance of the system. Recently,

other sources of information, such as temporal dynamics or implicit feedback, have been integrated into these models in order to improve accuracy [KOREN & BELL, 2011].

2.3 Social Recommenders

Recommender systems that exploit social context have been developed in recent years. Commonly, algorithms dealing with social context attempt to exploit the social connections of an active user. For example, [SHEPITSEN et al., 2008] employs a personalization algorithm for recommendation in folksonomies that relies on hierarchical tag clusters, which are used to recommend the most similar items to the user's closest cluster, by using the cosine similarity measure. Other works focus on graph based techniques for finding the most relevant items for a particular user, inspired by algorithms from quite different areas, successfully bringing them to social recommendation [KONSTAS et al., 2009]. Methods have been proposed in this context to deal with data sparsity and poor prediction accuracy by means of a factor analysis approach based on probabilistic matrix factorization, employing both the users' social network information and rating records [MA et al., 2008]. In this paper, we first focus on simple algorithms that only exploit social information (from now on referred to as "pure" social recommenders), and afterwards (Section 2.4) we study more general hybrid approaches that combine social information with explicit ratings, social tags, demographic information, etc.

2.3.1 Pure Social Recommender

Inspired by the approach presented in [LIU & LEE, 2010], we propose a pure social recommender that incorporates social information into the user-based CF model. This social recommender makes use of the same formula as the user-based CF technique (equation 1), but replaces the set of nearest neighbors with the active user's (explicit) friends. That is:

$$B[u_m, k] = B[u_m] = \{v \in \mathcal{U}: v \text{ is friend of } u_m\} \quad (7)$$

2.3.2 Friends Popularity Recommender

Similarly to the recommender proposed in [BARMAN & DABEER, 2010], and extended in the previous section (equations 5 and 6), we propose a social version of such recommender with the same rationale. We have implemented a friends' popularity recommender, where the algorithm suggests those items that are more popular among her set of friends, i.e., instead of using equation 5, we redefine the γ value as follows:

$$\gamma(u_m, i) = |\{v \in \mathcal{U}: v \text{ is friend of } u_m \text{ and } rat(v, i) \neq \phi\}| \quad (8)$$

Again, the scores predicted by this recommender cannot be interpreted as ratings, but can be used for generating item rankings. Because of its simplicity, we take this recommender as a baseline for the social-based recommender algorithms.

2.3.3 Personal Social Recommender

The approach of [BEN-SHIMON et al., 2007] explicitly introduces distances between users in the social graph in the scoring formula:

$$s(u_m, i_n) = \sum_{v \in X(u_m, L)} K^{-d(u_m, v)} rat(v, i_n) \quad (9)$$

For this recommender, the authors propose to use the Breadth-First Search algorithm in order to build a social tree for each user; this tree for user u is denoted as $X(u, L)$, where L is the maximum number of levels taken into consideration in the algorithm, and K is an attenuation coefficient of the social network that determines the extent of the effect of $d(u, v)$, that is, the impact of the distance between two users in the social graph (for instance, by using an algorithm that computes the distance between two nodes in a graph, such as Dijkstra's algorithm [DIJKSTRA, 1959]). Hence, when $K = 1$, the impact is constant and the resulting ranking is sorted by the popularity of the items. Furthermore, for this value of K, no expansion is employed and only directly connected users are involved in the score computation.

In this work, we use this recommender to obtain the raw scores in order to generate item rankings, since, like in the previous recommender, the scores calculated in this way cannot be interpreted as ratings.

2.4 Hybrid Recommenders

In addition to the recommenders presented in the previous section, there are other more complex algorithms that fall into the hybrid recommender category, and also exploit the social context of users. [BURKE, 2002] presents a detailed taxonomy where hybrid approaches are classified: *meta-level hybrid recommenders*, where the model learned by one recommendation technique is used as the input for another; *cascade hybrid recommenders*, in which recommendation is performed as a sequential process using different recommendation techniques; *weighted hybrid recommenders*, where scores from the recommendation techniques are aggregated using a linear combination or voting schemes; and *switched hybrid recommenders*, which are special case of *weighted recommenders*, in which one recommendation technique is turned off whenever the other is turned on.

In this paper, we explore different techniques in order to combine social and collaborative filtering techniques, as well as other strategies like Random Walks [KONSTAS et al., 2009], and extensions thereof. These techniques are useful not only for exploiting the social context of a user, but for providing higher coverage in extreme situations (such as the social or rating cold start, where no social context or ratings are available for a particular user).

2.4.1 Combined (Pure+CF) Recommender

Based on the recommender defined in Section 2.3.1, we propose a hybrid recommender that utilizes the user-based CF formula, but is based on all the active user's friends, as well as her most similar nearest neighbors, combining them into a new neighbor set:

$$B[u_m, k] = \{v \in \mathcal{U}: v \text{ is friend of } u_m\} \cup \{v \in B[u_m, k]\} \quad (10)$$

This new neighborhood is then used in equation 1 for estimating the score between a user and an item. Thus, this recommender integrates social and collaborative information in a simple but flexible way. According to Burke's taxonomy, it can be categorized as a *feature combination* hybrid, where the social information is treated as additional data for building the collaborative neighborhoods.

2.4.2 Random Walk-based Recommenders

Markov chains are stochastic processes that link states in a graph with certain transition probabilities, such that the probability of reaching a state depends only on the previous state, and no others. The matrix representing these probabilities for each pair of states is called the transition probability matrix. The random trajectory described in such a state graph based on the transition probabilities is called a Random Walk (RW).

The hypothesis behind an RW applied to social graphs is that users, who have evaluated the same items, probably have similar tastes and, as a statistical trend, will be connected by, comparatively, a large number of short paths. Users with different tastes would be expected to be connected by a smaller number of paths in the social network, and, therefore, the shortest paths between them should be longer. The notion of path length can also consider weights assigned to the edges connecting users in the adjacency matrix of the social graph.

In the following, we describe the basics of this approach, as well as other possible extensions. These recommenders can fit, again, as *feature combination* hybrids, since a generic model (Random Walk) is built using input from many different sources, such as friendship links, preferred items, and even tags (not considered in this work).

For readability, we show here the final equations for the definition of the adjacency matrix $\mathcal{G}$ here, but leave the detailed derivation for Appendix A. Given a feature F_k available in the collection, and its associated enumerated set image E_{F_k}, the elements of this matrix $g_{F_k}(\cdot,\cdot)$ may be defined using two alternative formulations, depending on whether explicit (equation 11) or implicit (equation 12) information is available:

$$\begin{aligned} g_{F_k}: \mathcal{U} \times \mathcal{I} &\longrightarrow E_{F_k} \\ (u,i) &\longrightarrow g_{F_k}(u,i) \end{aligned} \tag{11}$$

$$\begin{aligned} g_{F_k}: \mathcal{U} \times \mathcal{I} \times \{0,1\}^{|\mathcal{F}|} &\longrightarrow \mathbb{R} \\ (u,i,\vec{\alpha}) &\longrightarrow g_{F_k}(u,i,\vec{\alpha}) \end{aligned} \tag{12}$$

Once the matrix $\mathcal{G}$ has been defined, it is possible to compute the recommendations by means of a Random Walk based recommendation algorithm [KONSTAS et al., 2009], and some of its variations. In the application to RW algorithms to recommendation, users and items are considered as the nodes of the state graph. The nodes are linked to each other by a probability $p_{n,m}$ of going from node $n(t)$ at some state or time t to the adjacent node m at the next state, i.e., $m(t+1)$. These probabilities define the transition matrix *P*, and are calculated based on the values of the matrix $\mathcal{G}$ as follows:

$$p_{n,m} = P(m|n) = \frac{g_{F_k}(n,m)}{\sum_{d \in |D|} g_{F_k}(n,d)} \tag{13}$$

where *D* is the set of nodes of the graph. Besides users and items, other components can be included in the state graph, such as tags, genres, etc, resulting in further variations of the basic RW recommendation approach. Based on the above matrices, the probability distribution of states at a given time *t* associated to the RW is defined by:

$$\Pi(t) = \left[\pi_1(t), \pi_2(t), \cdots, \pi_{|D|}(t)\right]' \tag{14}$$

where $\pi_n(t) = \left[P(n(t)|m_1), \cdots, P\left(n(t)|m_{|D|}\right)\right]$ represents the probability vector of being at node n at time t. The fact that $\pi_n(t)$ depends only on the states at time $t-1$, as determined by transition probabilities, provides the basis for recursive definition of $\Pi(t)$:

$$\begin{aligned} \Pi(t+1) &= P^T \Pi(t) \\ \Pi(0) &= \Pi^0 \end{aligned} \tag{15}$$

where Π^0 is the identity matrix.

Random Walk with Restarts (RWR) introduces a probability of restarting from the original state, no matter what the current state is. This is formulated by modifying how the state transition probabilities are computed, in this case:

$$\Pi(t+1) = (1-r)P^T\Pi(t) + rQ$$
$$Q = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \iddots & \vdots \\ \vdots & \iddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \quad (16)$$

where Q is the identity matrix and the probability of restarting is represented as r. Therefore, the corresponding vector $q' = Q_{\cdot i}$ according to the i-th column in the RWR approach would be $q' = \left[0, \cdots, 0, \underbrace{1}_{i}, 0, \cdots, 0\right]$, with 1's only in the i-th position. Moreover, if $r = 0$, we obtain the standard RW. In fact, the higher the value r is, the higher is the probability to restart at the departure state, i.e., in that case, $\Pi(t+1) \sim Q$. The personalization matrix Q defined in this way allows to use personal user tastes during the random path, since a bias towards the starting node is introduced by using this matrix.

In [KONSTAS et al., 2009], it is shown that the performance of a recommender system using the RWR method is improved when the extra knowledge provided by the users' social activity is used. Because of that, we propose a modification in the way the matrix Q is computed, by including all other users who also evaluated the same item. In our approach, each component of Q is computed by means of the following expression:

$$\forall g_{F_k}(a,b) \in \mathcal{G}; Q_{ab} = \frac{\delta_{a,b}}{\sum_{\tau \in \{1,\ldots,Dim(\mathcal{G})\}} \delta_{\tau,b}}; \delta_{a,b} = \begin{cases} 1, & g_{F_k}(a,b) > 0 \\ 0, & g_{F_k}(a,b) = 0 \end{cases} \quad (17)$$

We denote this approach as Personalized Random Walk with Restart (PRWR). Using the Personalized approach, and assuming that β elements in each column i have a non zero value (i.e., $\sum_{\tau \in \{1,\ldots,Dim(\mathcal{G})\}} \delta_{\tau,i} = \beta$), the resulting vector would be

$$q' = Q_{\cdot i} = \left[\frac{1}{\beta}, \cdots, 0, \frac{1}{\beta}, \frac{1}{\beta}, \cdots, 0\right] \quad (18)$$

Thus, we introduce the probability to restart at some state as a uniform weight for each user u taking into account their relative importance in the user set. The assigned values distribute the restart probability uniformly among all the users who have evaluated the item, as opposed to just the target user.

3. EXPERIMENTAL COMPARISON

We report in this section the results obtained in empiric observations with the different recommenders described in the previous section. We shall first describe the experimental methodology, and then, the details and results of the conducted experiments, which encompass different considerations regarding the test set. In particular, firstly, we have used a test set containing only users with social context (social evaluation) and, secondly,

a test set where half of the users have an empty social context, and the other half have a non-empty one (collaborative-social evaluation).

3.1 Experimental Setup

This section shows the specific details about the dataset used in our experiments, the different metrics included in the study, and the chosen configurations for the involved recommenders.

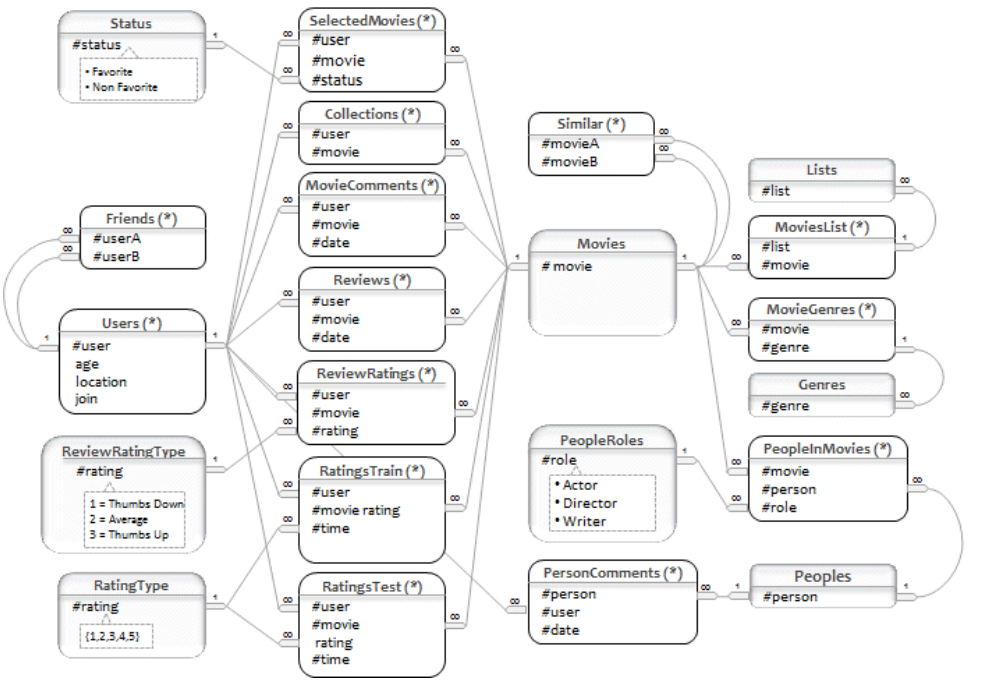
3.1.1 Dataset

The experiments we report here are conducted on the Filmtipset dataset provided for the CAMRa Challenge [SAID et al., 2010]. Filmtipset is Sweden's largest online social community in the movie domain, with more than 80,000 users. It contains a wide range of user and movie information that could be exploited to make predictions of users' tastes and provide movie recommendations.

The Filmtipset dataset contains explicit information about movies, such as film genres and featuring actors. It also contains personal information about the users, such as their social contacts and their movie reviews. Finally, it also has implicit information like the frequencies with which a user watches a film of a specific genre, or makes reviews of a specific person. The CAMRa dataset is quite unique in its kind, as a publicly available resource, in the range and variety of user data provided. In particular, to the best of our knowledge, it is as of today among the only three public datasets (see Section 4) including both user preference data and social relationships, thus, meeting the requirements for the experimental work in our research. Of all the available data in this dataset –different types of reviews, comments, favorites, etc – we focus in our experiments on the exploitation of explicit preferences and social links, as the most general and representative type of data in similar information systems.

Figure 1 shows the entity/relationship model of the Filmtipset dataset. The tables marked with an asterisk (*) are those provided in the original dataset. Additionally, we have included the following tables: Status, ReviewRatingType, RatingType, Movies, People, PeopleRoles, Lists, and Genres, to generate a database with referential integrity to optimize queries in the computation process. We note that, in order to create the Movies table, we take into account existing relationships with all those tables of the Filmtipset dataset containing movies, as shown in Figure 1. That is, the Movies table includes information from all tables but Friends, Users, and PersonComments tables. We proceed in the same way to create the People, Genres, and Lists tables.

Figure 1. The entity/relationship model of the CAMRa 2010 Filmtipset Social Recommendation dataset. Tables with (*) are those provided with the dataset.

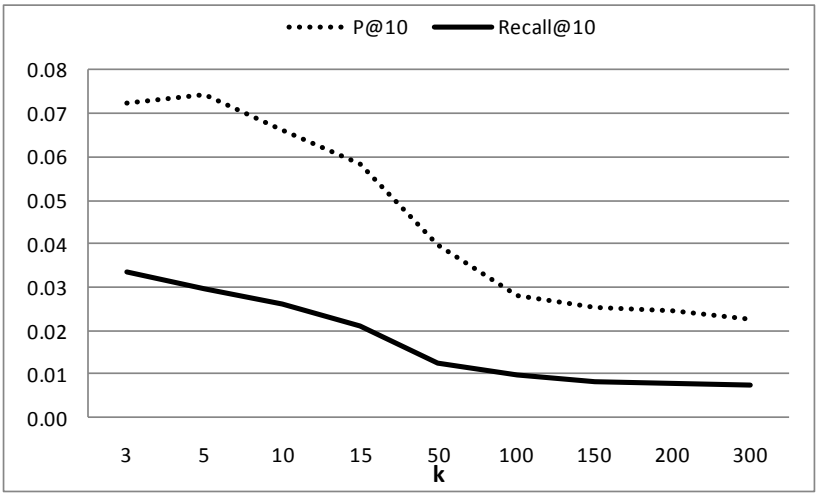


With the pre-processed data, we build and evaluate the recommenders explained in Section 2 using the metrics and parameters specified in the following sections.

3.1.2 Metrics and Methodology

We compute several standard accuracy metrics from the Information Retrieval field, which are being increasingly adopted by the Recommender Systems research community: Precision, Recall [BAEZA-YATES & RIBEIRO-NETO, 1999], and Normalized Discounted Cumulative Gain (NDCG) [JÄRVELIN & KEKÄLÄINEN, 2002] at top positions 5, 10 and 50. These evaluation metrics capture the accuracy in matching the user's interests when she is presented with an item ranking. The chosen ranking cutoff positions are standard in common practice in IR. They are suitable in the RS context as well, where it seems sensible to assume that the top positions of recommendations are more frequently examined by users, and items beyond, e.g., position 50, are rarely accessed in common real scenarios [WANG, HAWK, TENOPIR, 2000; JANSEN & SPINK, 2003; CARTERETTE, 2011]. Note that, in contrast with the typical experimental setup in Information Retrieval, Recommender Systems' false positives might be overestimated, since the items not rated by the user are considered as non relevant, where in reality, we may only say the preference of the user for such items has not been observed in the system –i.e., their relevance or non-relevance is unknown to the system,

Figure 2. Performance evolution of user-based method depending on the neighborhood size (k).



when in fact, those items could be good and serendipitous recommendations. For this reason, we report recall along with precision. To compute the measures listed above we used *trec_eval*[3], a public program distributed in the scope of the Text Retrieval Conference (TREC) to evaluate retrieval system results by using the standard NIST procedures.

The evaluation methodology we used in our experiments is the following: for each user, a recommender is requested to provide a preference score for every item in the test set, except for those items already included in the user's (training) profile; from these scores, a ranking is generated, the items are sorted in descending preference score order. In Appendix B, a detailed analysis of the number of generated ties by this strategy for each recommender is provided.

3.1.3 Configuration of the Recommenders

We evaluated different neighborhoods for the user-based recommender, ranging from 3 to 300. For the sake of clarity, in each situation, we have selected the best performing one –by which we mean the one with highest performance with respect to more metrics– and denoted it as *UB*. For comparison purposes, Figure 2 summarizes the performance results obtained for each neighborhood size for precision and recall at 10 –a similar trend is observed for other cutoffs and performance metrics. In contrast with previous experiments reported using error-based metrics such as [HERLOCKER, KONSTAN, RIEDL, 2002], in this situation, smaller neighborhoods seem to provide better accuracy

[3] http://trec.nist.gov/trec_eval/

(higher precision and recall) than larger ones. Thus, in the rest of this section, a neighborhood size of 5 will be used.

Friends popularity recommender is taken as a baseline and named as *FriendsPop*, the pure social recommender is denoted as *PureSocial*, the similar users popularity recommender as *SimPop*, and the personal social recommender *Personal*. The latter recommender is evaluated using an attenuation coefficient of $K = 2$ and a level $L = 6$ (see equation 9). Additionally, the matrix factorization method *SVD* is also included in the experiments, where the dimension of the latent space varies from 50 to 200.

For this experiment, we also tested different implementations of random-walk algorithms (see Section 2.4) according to our previous work [DIEZ et al, 2010]. More specifically, the random walk method was built using the set of features $\mathcal{F} = \{F_1, F_2, F_3\}$, where F_1 is the relation defined by the reviews received by each item; F_2 is the relation defined by the ratings given to the items in the collection; and F_3 is the binary relation defined as whether a user has indicated an item as favorite or not. According to our previous work, the best performing RW was the one involving the feature F_1, i.e., reviews, with the vector $\vec{\alpha} = \{1,1,1\}$, and is denoted as *RW*. The best performing random walk method with restarts employs a restart probability of 0.3, and uses also the feature F_1 and $\vec{\alpha} = \{1,1,1\}$ parameters; we denote it as *RWR*. Finally, the personalized version is denoted as *PRWR*, which makes use of the same feature and weights than the previous models, but using the personalized matrix defined in equation 17.

The other hybrid is denoted as *Combined* where, as with the user-based method, different neighborhood sizes have been considered in addition to the user's friends, and only the best performing one is reported.

3.2 Social Evaluation

Table I shows the performance results obtained by the recommenders described in Section 2 for the test set provided in the social track of the CAMRa challenge [SAID et al., 2010]. As explained previously, in the test set provided in the dataset, all users have a non-empty set of contacts –at least two friends. Later on we will extend our study to a situation where the social network coverage is only partial.

As expected, the baseline for the social-based recommenders (friends' popularity recommender) is outperformed by the pure social recommender. However, the analogous collaborative recommender *SimPop* outperforms *SVD* in terms of precision, but underperforms in terms of recall. Nevertheless, *SimPop* always performs better than its social counterpart (*FriendsPop*).

Personal recommender is the best performing algorithm under different metrics, despite its simplicity. We presume this behavior is due to the relatively high value of *L*, which causes the social tree of every user to spread along several nodes.

Results from the best performing random-walk algorithms according to previous work [DIEZ et al, 2010] are also included in the table. RW algorithms perform very well in comparison with other social-based recommenders (leaving aside the personal social recommender). However, there is no clear winner among them, since depending on the evaluation metric, *RW* performs better than *RWR*, or vice versa.

The performance of the evaluated hybrid approaches varies dramatically: while *Combined* is able to achieve better performance than *UB* and *PureSocial* for many different metrics, there are other situations where it obtains worse performance than their corresponding baselines.

Table I. Summary of the results for the social evaluation (i.e., original challenge test set). Best recommenders in bold. Best recommenders for each type, in italics.

		P@			**Recall@**			**NDCG@**		
	Model	**5**	**10**	**50**	**5**	**10**	**50**	**5**	**10**	**50**
User Based	UB	*0.085*	*0.072*	*0.045*	*0.022*	*0.034*	*0.099*	*0.088*	*0.083*	*0.082*
	SimPop	0.062	0.063	0.041	0.005	0.011	0.061	0.066	0.068	0.070
	SVD	0.044	0.038	0.033	0.007	0.015	0.054	0.044	0.041	0.048
Social Based	PureSocial	0.062	0.057	0.053	0.015	0.032	0.149	0.064	0.064	0.094
	FriendsPop	0.001	0.001	0.004	0.000	0.000	0.033	0.001	0.001	0.011
	Personal	***0.370***	***0.344***	***0.233***	***0.143***	***0.226***	***0.513***	***0.408***	***0.414***	***0.449***
Hybrid	RW	0.080	0.070	0.066	0.013	0.024	0.107	0.082	0.077	0.095
	RWR	0.063	0.056	0.057	0.016	0.029	0.110	0.068	0.066	0.090
	PRWR	0.043	0.044	0.059	0.007	0.017	0.122	0.047	0.048	0.085
	Combined	*0.084*	*0.077*	*0.073*	*0.017*	*0.035*	*0.168*	*0.086*	*0.085*	*0.119*

We observed that some of these algorithms are sensitive to the minimum number of users who have evaluated the current item in order to compute the predicted score. Some of the recommenders may produce their predictions using a very small number of users, and, thus, decrease their performance. Both collaborative algorithms and social-based algorithms can be sensitive to this parameter which, from now on, we will refer to as **overlap threshold value**. It is defined as the minimum number of neighbors which should have rated the current item in order to consider the current recommendation as valid. Table II shows the results when this threshold is set to 2, i.e., whereas Table I shows results where only one neighbor is required to have evaluated the item the

prediction aims at, Table II shows results where at least two neighbors are required. In this setting, we obtain a significant improvement for all the algorithms. *FriendsPop* and RW recommenders are not included in this table because they do not include such a constraint in their definitions, although we plan to include it in the future.

Table II. Summary of the results for the social evaluation when overlap threshold is 2. Best recommenders in bold. Best recommenders for each type, in italics.

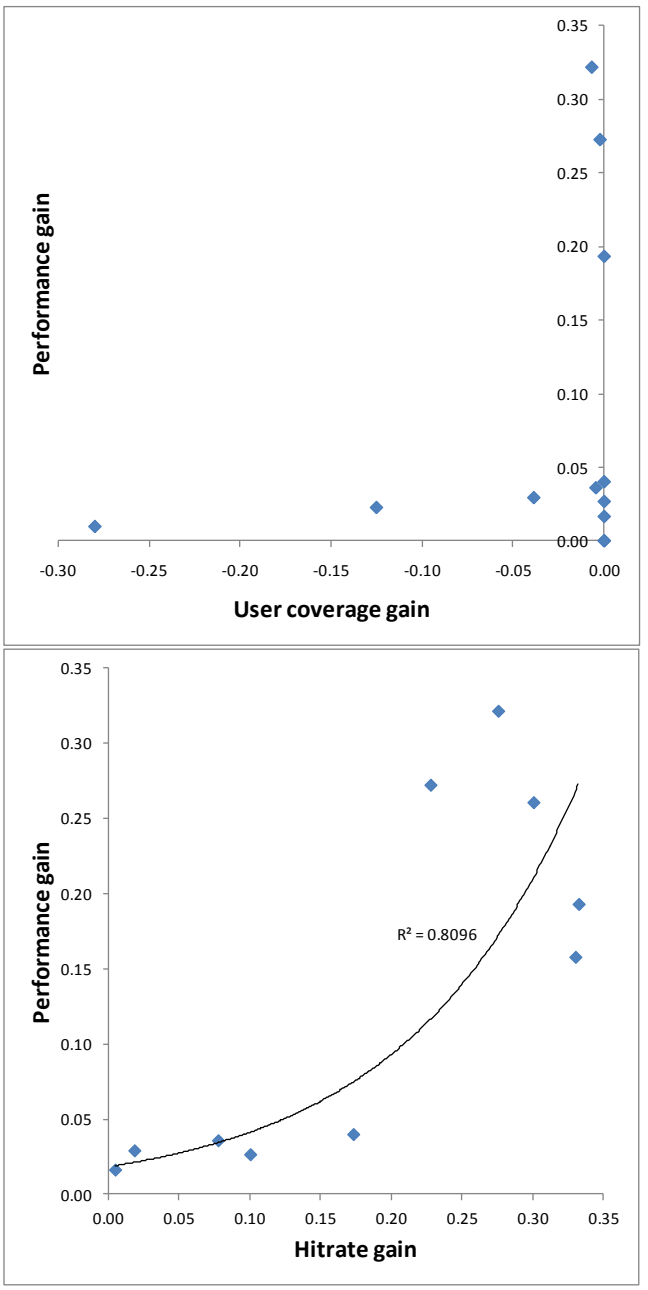
		P@			**Recall@**			**NDCG@**		
	Model	**5**	**10**	**50**	**5**	**10**	**50**	**5**	**10**	**50**
User Based	UB	*0.110*	*0.092*	0.042	*0.027*	*0.044*	0.099	*0.117*	*0.108*	0.092
	SimPop	0.062	0.063	0.041	0.010	0.023	0.082	0.066	0.068	0.070
Social Based	PureSocial	0.277	0.257	0.216	0.134	0.214	0.535	0.309	0.321	0.415
	FriendsPop	0.001	0.001	0.004	0.000	0.000	0.033	0.001	0.001	0.011
	Personal	***0.374***	***0.347***	***0.238***	**0.151**	**0.237**	**0.542**	***0.414***	***0.422***	***0.466***
Hybrid	Combined	*0.274*	*0.269*	*0.203*	*0.114*	*0.210*	*0.497*	*0.300*	*0.324*	*0.391*

This gain in performance, however, is not obtained without cost. Figure 3 shows the loss in **user coverage** suffered by some algorithms when the overlap threshold is increased. We define the user coverage as the amount of users for which the system is able to produce a recommendation. This concept is related to the prediction coverage described in [HERLOCKER et al., 2004], but focused on users instead of items. User coverage, in contrast with item (or catalog) coverage, may have a large impact when comparing different recommenders' performance, since some methods may obtain better accuracy at the expense of reducing the proportion of users able to receive recommendations. For instance, in a typical recommendation scenario, the user coverage should be 100%, but since recommender systems typically deal with very sparse data, it may be the case that some users cannot be recommended any item.

In our experiments, we observe that, except in a few cases (where coverage loss is zero, and the improvement is positive), the rest of the recommenders either lose coverage while gaining performance, or improve nothing at all; there is, however, no clear relation between both variables. This observation reinforces the one presented in [HERLOCKER et al., 2004], where the authors claim that "coverage must be measured in combination with accuracy" in order to avoid strange behaviors in recommendation. Item coverage was also measured in the experiments, and similar results were obtained. An additional measure, *hitrate*, is also included in the figure. It is computed as the percentage of users with at least one correct recommendation. Note that this measure is equivalent to the *success* metric, as defined in TREC; besides, this metric requires an additional parameter,

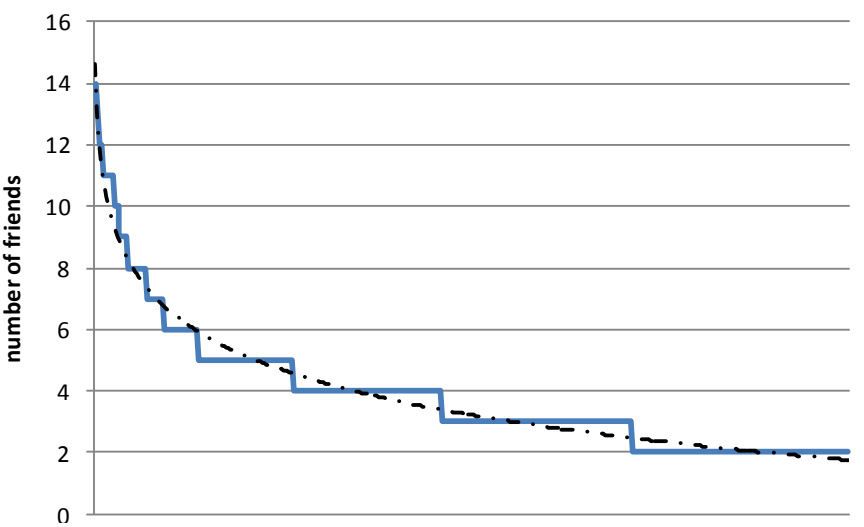
namely the maximum number of documents to look up for the correct recommendation. In our experiments, we set this parameter to 50, since we compare these values against nDCG@50. We can observe in the figure how this metric provides an adequate balance between coverage and performance, since the better the performance, the more likely this metric obtains a higher value ($R^2 = 0.8096$, determination coefficient for an exponential fit).

Figure 3. Comparison of performance gain (NDCG@50) versus user coverage and hitrate gain.



The design of the CAMRa dataset, and more specifically, the selection of test users as provided in the challenge's social track in which the above results are observed, is representative of online applications in which every target user has a non-empty list of contacts (see Figure 4). This is the case of social-centric systems such as Facebook, Linkedin or Twitter, but in many social media applications, such as Delicious or Last.fm, the social network coverage is only partial –some users use it, some do not. In fact, the Filmtipset dataset belongs to this case: considering the set of all users, more than 10,000 out of about 16,500 users do not have any friends in the system. The number of contacts per user follows a power law distribution, where the average number of friends per user is 0.74 and the mode among users (with at least one friend) is 1. If we take this dataset as representative of social media systems, the presence of contacts by itself does not guarantee the accuracy of social recommendation, and intermediate cases, where social data is available but not enough to support optimal recommendation, would rather seem to be the norm. We, therefore, consider such common real situations in our study, where there is a mixed degree of participation in social networking. For this purpose, we simulate an alternative scenario by adding an equal amount of users without friends to the test set, as we describe next.

Figure 4. Friends distributions among the users composing the original test set of the social track. Note how a logarithmic regression line fits almost perfectly the distribution. The total number of users is 439, the maximum number of friends is 14 and the minimum is 2.



3.3 Collaborative-social Evaluation

For our second experiment, we add an equal number of test users as was included in the original test set (i.e., 439 users), randomly sampled among users having no friends. We also keep the same ratio of test ratings per user as in the original data split. Results for

this new test set are shown in Tables III and IV. Like in the previous evaluation, we can observe that using a higher overlap threshold improves the performance of all the recommenders. In general, the relative performance of the recommenders is quite similar to that presented in Section 3.2, since for some metrics the user-based recommender outperforms *PureSocial* when no threshold is set, whereas *PureSocial* increases its performance notably when this threshold is increased. Once again, *Personal* is the best performing algorithm in both situations.

The behavior of hybrid recommenders, however, is substantially different: Random Walk recommenders no longer outperform simple social-based recommenders; and *Combined* obtains better performance than both their baselines (*UB* and *PureSocial*) when no threshold is used. When a higher overlap threshold is used, combined recommenders do not get as close to the baselines as in the other evaluation.

Table III. Summary of the results for the collaborative-social evaluation and an overlap threshold of 1. Best recommenders in bold. Best recommenders for each type, in italics.

		P@			**Recall@**			**NDCG@**		
	Model	**5**	**10**	**50**	**5**	**10**	**50**	**5**	**10**	**50**
User Based	UB	*0.054*	*0.052*	*0.036*	*0.021*	*0.036*	*0.126*	*0.060*	*0.062*	*0.080*
	SimPop	0.042	0.042	0.030	0.011	0.025	0.100	0.043	0.046	0.060
	SVD	0.028	0.026	0.020	0.012	0.017	0.050	0.029	0.030	0.038
Social Based	PureSocial	0.055	0.050	0.047	0.013	0.029	0.132	0.053	0.054	0.082
	FriendsPop	0.001	0.001	0.004	0.000	0.000	0.028	0.001	0.001	0.009
	Personal	***0.369***	***0.342***	***0.230***	**0.139**	**0.222**	**0.510**	***0.403***	***0.409***	***0.443***
Hybrid	RW	0.023	0.021	0.030	0.004	0.008	0.078	0.026	0.025	0.048
	RWR	0.023	0.021	0.030	0.004	0.008	0.078	0.026	0.025	0.048
	PRWR	0.023	0.021	0.030	0.004	0.008	0.078	0.026	0.025	0.048
	Combined	*0.055*	*0.053*	*0.049*	0.008	0.012	0.080	*0.059*	*0.062*	*0.097*

Table IV. Summary of the results for the collaborative-social evaluation for an overlap threshold of 2. Best recommenders in bold. Best recommenders for each type, in italics.

		P@			**Recall@**			**NDCG@**		
	Model	**5**	**10**	**50**	**5**	**10**	**50**	**5**	**10**	**50**
User Based	UB	*0.090*	*0.073*	0.029	*0.045*	*0.065*	0.100	*0.103*	*0.099*	0.088
	SimPop	0.042	0.042	0.030	0.011	0.025	0.100	0.043	0.046	0.060
Social Based	PureSocial	0.270	0.249	0.209	0.132	0.209	0.523	0.298	0.310	0.403
	FriendsPop	0.001	0.001	0.004	0.000	0.000	0.028	0.001	0.001	0.009
	Personal	***0.372***	***0.345***	***0.235***	**0.147**	**0.235**	**0.538**	***0.409***	***0.417***	***0.460***
Hybrid	Combined	*0.185*	*0.173*	*0.121*	*0.093*	*0.154*	*0.331*	*0.207*	*0.218*	*0.258*

Coverage analysis explains the different performance of social and hybrid recommenders. Figures 5a and 5b show the relation between precision and coverage in the two evaluations conducted, first with respect to the user coverage metric (Figure 5a), and then with respect to the hitrate metric (Figure 5b). In both figures, we group the recommenders according to their source of information: social (*PureSocial*, *FriendsPop*, *Personal*), collaborative (*UB*, *SimPop*, *SVD*) and hybrid (*RW*, *RWR*, *PRWR*, *Combined*). We also show the performance of the recommenders for the two analyzed overlap thresholds (filled vs. unfilled markers).

Figure 5a. Comparison between user coverage and performance (NDCG@50) for the different recommenders used in the experiments grouped by its nature: social-based, collaborative-based, and hybrids. Unfilled markers represent that the overlap threshold has been increased. Top figure shows the comparison using the original test set, and the bottom one uses the new test set (collaborative-social evaluation). Coverage ratio has been normalized for comparison purposes.

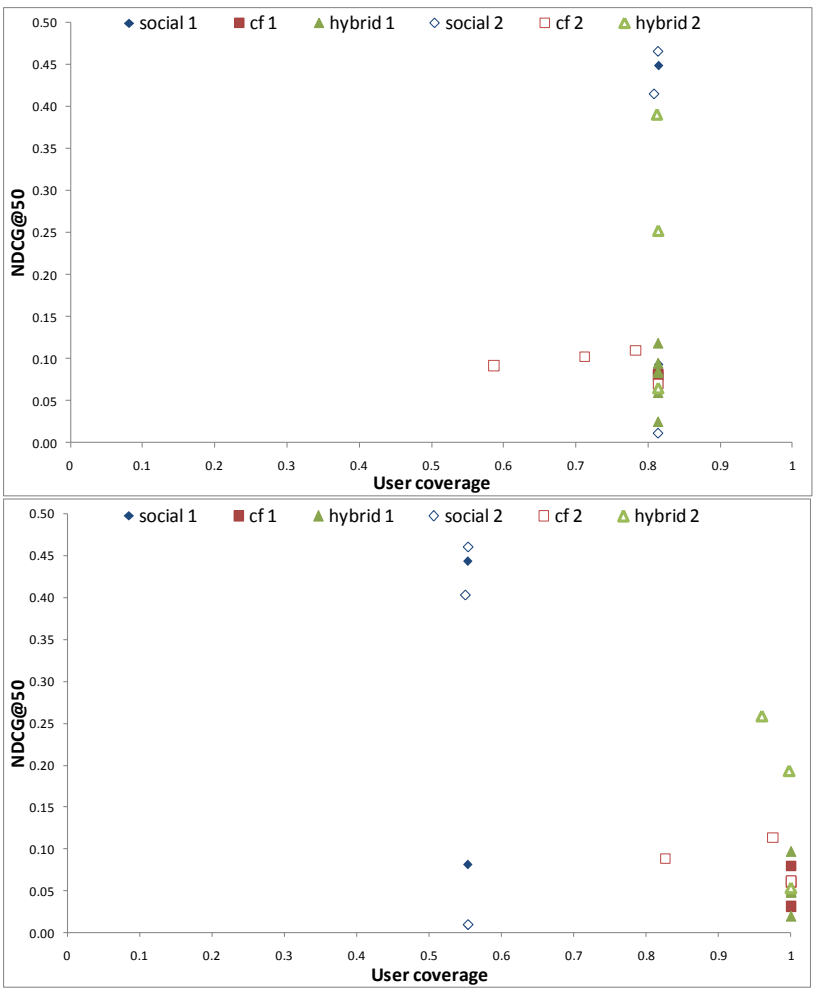
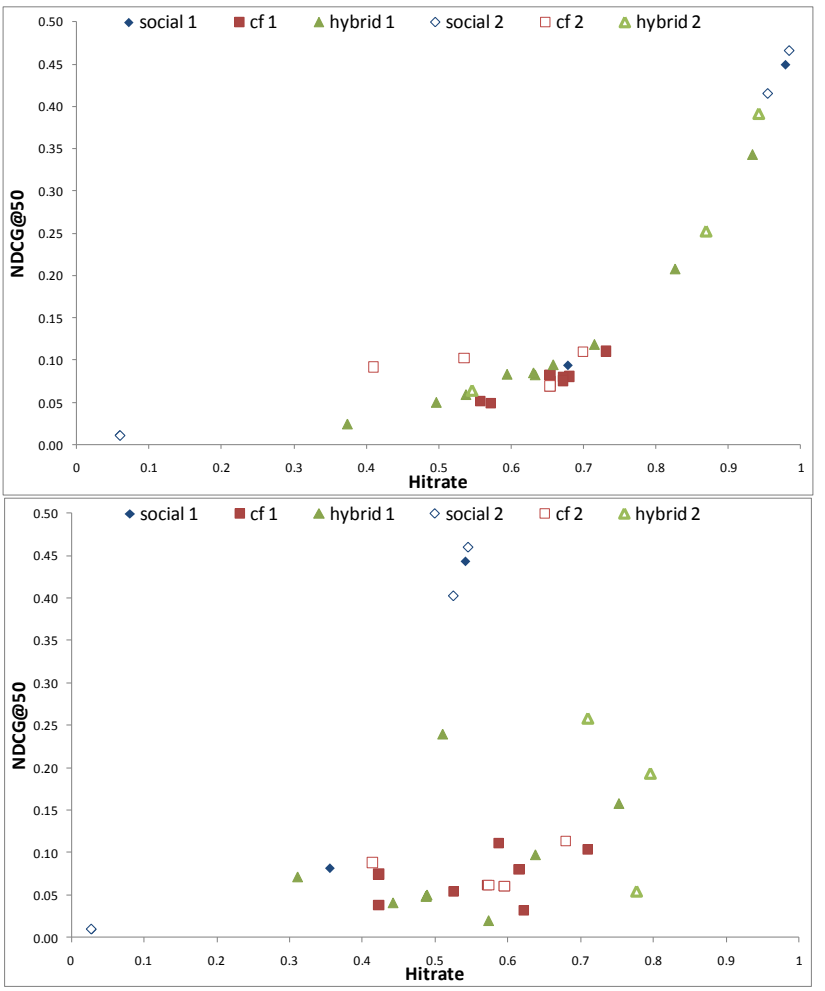


Figure 5a shows the shift in performance at the expense of coverage when changing the overlap threshold, since unfilled markers tend to be leftwards (less coverage) or

higher (more performance) than their corresponding filled markers in the two evaluations performed. We can also observe that, in the social evaluation (top figure), most of the recommenders obtain the same user coverage, whereas in the second evaluation, social recommenders have less coverage, although their performance is higher than other recommendation methods.

Figure 5b presents the coverage results with respect to the hitrate measure. In this case, social and hybrid recommenders obtain higher values for the social evaluation. Then, in the collaborative-social evaluation, hitrate values decrease for most of the recommenders, leading to consistent results with respect to those found using the user coverage metric.

Figure 5b. Comparison between hitrate and performance (NDCG@50) for the different recommenders used in the experiments grouped by its nature. See Figure 5a for notation.



Thus, the collaborative-social evaluation confirms the coverage degradation of pure social methods as the social network gets sparser, as one might expect, providing a quantitative assessment of the extent of the loss. Furthermore, the results in Figures 5a and 5b confirm the advantages expected for hybrid approaches: they are the best performing algorithms among those having high user coverage and hitrate. This observation motivated further analysis and research towards recommender hybridizations aimed at maximizing coverage and performance, which we address in the next section.

3.4 Evaluation of Hybrid Recommenders

When aiming to achieve the most accurate performance, while preserving the highest coverage, it would seem that in the original evaluation there was no reason for hybridization, since all the algorithms had similar coverage, and social-based recommenders were the best performing. Thus, in that case, the answer to our problem would be simple: just use social-based recommenders. This conclusion, however, arises from a partial experiment, as we explained in the previous section: by design, the test set favors algorithms using social information. To address that problem, in the second evaluation, we presented a scenario where hybrid algorithms may prove their usefulness: collaborative filtering recommenders achieved high coverage, and social-based recommenders performed well, although with low coverage, and indeed, some of the evaluated hybrid recommenders obtained good results –although not clearly superior to collaborative or social recommenders.

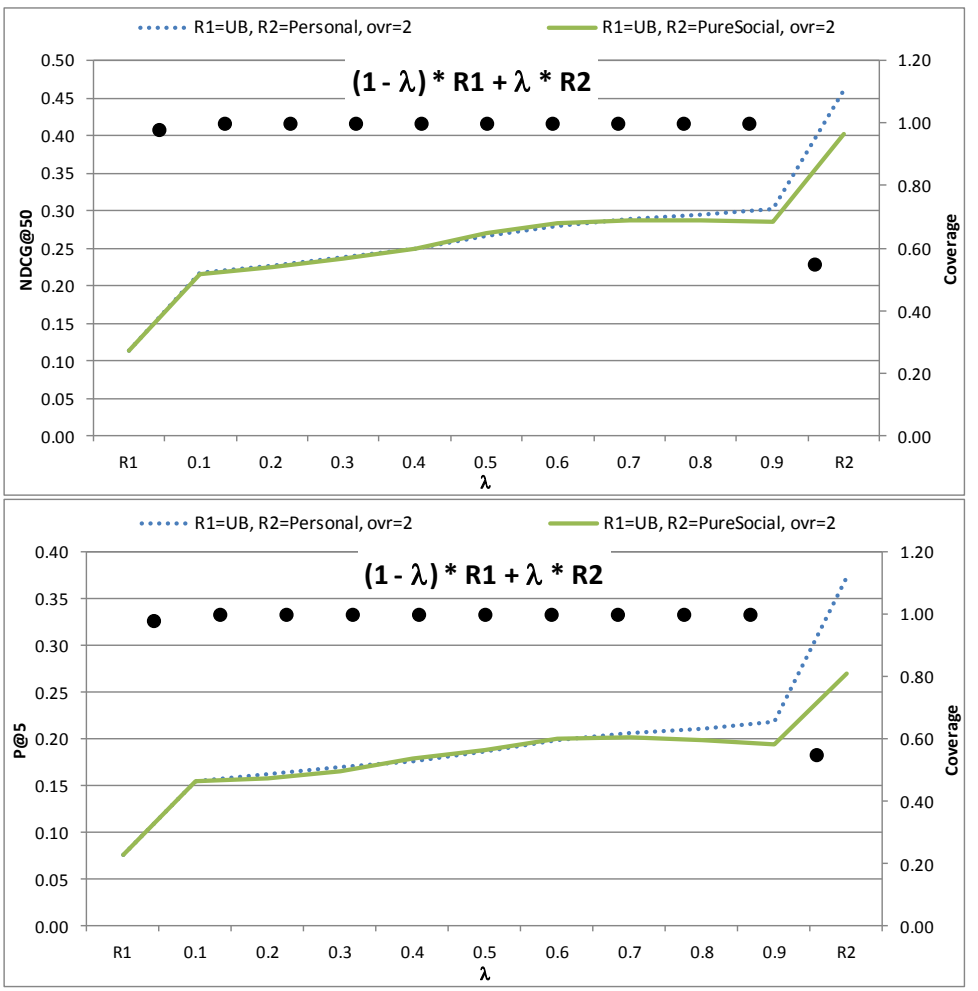
In this section, we analyze another hybridization, which combines multiple recommender systems to produce an output (score) [BURKE, 2002], in contrast to *Combined* or RW, where each recommender takes two or more different sources of information as input. Specifically, we focus on **weighted hybrid recommenders**, where the score of each technique is aggregated by a linear combination with a weight λ between 0 and 1 in the following manner:

$$s(u_m, i_n) = (1 - \lambda) \times s_1(u_m, i_n) + \lambda \times s_2(u_m, i_n) \qquad (19)$$

Except for λ = 0 and 1, a weighted hybrid algorithm achieves a non-zero coverage for the entire set of users for which each combined system is able to produce a recommendation. In general, the optimum weight is derived by examining the performance when using all possible weight values [BURKE, 2002]. Figure 6 shows the results for different values of λ, gradually increasing from 0 to 1 by increments of 0.1. In this figure, two combinations of recommenders are presented: H1 (*UB* and *Personal*, with an overlap threshold of 2) and H2 (*UB* and *PureSocial*, with the same threshold). An

additional hybrid was evaluated, namely, the same as H2, but with an overlap threshold of 1. As expected, the accuracy for this hybrid is lower and, because of that, its performance is always inferior to H2. Hence, we omit this result from our analysis. In summary, these results confirm our previous claim: the gain in coverage (and performance) when combining collaborative and social information is clear.

Figure 6. Performance of the different static hybrid recommenders evaluated. Top figure shows NDCG@50, and bottom figure shows P@5. Black dots represent the coverage obtained at each recommender combination.



Both presented hybrids obtain maximum coverage, since one of the combined recommenders in every situation is a user-based recommender. The other recommenders in the combination (*PureSocial* and *Personal*) have been chosen because they are purely social-based recommenders, and have proven to perform very well, although not achieving high coverage.

3.5 Dynamic Hybridization

The type of hybridization presented above weights the recommenders in a static way. That is, once the value of λ is fixed, recommendations from each technique receive the same weight, independently of the target user. In this section, we propose to make **dynamic hybrid recommendations**: depending on some user characteristics or attributes, one of the combined recommenders is given more or less weight. Therefore, in this case, the value λ is fixed for each user rather than for the entire population, aiming to promote the recommender that is expected to perform best for each particular user, or equivalently $\lambda = \lambda(u_m)$. Hence, equation 19 can be rewritten as

$$s(u_m, i_n) = \big(1 - \lambda(u_m)\big) \times s_1(u_m, i_n) + \lambda(u_m) \times s_2(u_m, i_n) \qquad (20)$$

The user dependent weight $\lambda(u_m)$ would be computed as a function of some user attributes, such as her profile or behavior in the system, i.e., $\lambda(u_m) = \mathrm{f}(u_m)$. In this paper, since the hybrids in consideration consist of combinations of collaborative filtering and social-based recommenders, we propose to use graph-based measures as indicators of the user strength in the social network. The utilized indicators of the user strength in the community are: *user degree* (the number of friends in the social network) and *average neighbor degree* (the mean number of friends of each user's friend), *PageRank* and *HITS score* (measures of connectivity relevance within a social network) [BRIN & PAGE, 1998; KLEINBERG, 1999], *betweenness centrality* (an indicator of whether a user can reach others on relatively short paths) [FREEMAN, 1977], and *clustering coefficient* (probability that the user's friends are friends themselves) [WATTS & STROGATZ, 1998].

More specifically, we apply these measures to set the weight of the social recommender in the hybrid combination, in such a way that when a user is very socially relevant (e.g., she has a lot of friends or plays a special role in her community), the recommendations from the social recommender are made more influential than those from the user-based CF recommender, and vice versa. That is, for each hybrid recommender, $\mathrm{f}(u_m)$ is set according to the (normalized) value of each of the above presented graph-based measures for that particular user.

Table V. Performance results of the two hybrid recommenders tested: H1 (*UB* and *Personal*, overlap threshold is 2), H2 (*UB* and *PureSocial*, overlap threshold is 2). The performance of the *dynamic hybrid* recommenders is included, as well as the one of the *best static* hybrid recommender and of the *static with weight 0.5*. The performance metrics are NDCG@50 and P@5. Improvements over the *best static* are denoted with bold font, and over the *static 0.5* with italics. Statistical significant ($p<0.05$) performance differences between *dynamic hybrid* recommender and *static 0.5*, *best static* and *both* static 0.5 and best static are respectively marked with *, † and ‡.

	NDCG@50		**P@5**	
Hybrid	**H1**	**H2**	**H1**	**H2**
Average neighbor degree	*0.301**	*0.280**	**0.219***	*0.199*
Centrality	*0.296**	*0.276*†	**0.222***	0.188
Clustering Coefficient	*0.294**	*0.274*†	*0.211**	0.188
Degree	**0.309**‡	*0.286**	**0.233**‡	*0.197*
HITS	**0.306***	*0.284**	**0.225***	*0.197*
PageRank	0.303*	**0.289***	**0.227**‡	**0.200***
Static 0.5	0.266	0.270	0.186	0.189
Best static	0.303	0.287	0.218	0.199

Table V shows the results obtained with the proposed dynamic hybridization technique. We compare the results from the dynamic hybrid recommendations against those of the best static weighted hybrid recommender (for a given evaluation metric), and a static weighted hybrid recommender with $\lambda=0.5$. This value is the natural choice (and best prior on average) in the absence of information about the recommender systems to be combined. It is also appropriate when the recommenders are not reliable or, in summary, there is uncertainty about the output of the recommenders. In the table, we may observe that, in general terms, **dynamic hybrids outperform the best static hybrid recommender in each situation**, or at least, they obtain a very similar performance. It is important to note that the best static one is different for each combination (namely, $\lambda=0.9$ for H1 and 0.8 for H2), which further highlights the importance of this result, in that the best static one is not actually a real configuration: the best static configuration would require manual tuning of λ, and even so, the table shows the *post-hoc* best static one, which a manually tuned λ would not even guarantee that such improvement is obtained. Nonetheless, although no dynamic method is substantially better in every situation, the PageRank indicator seems to be a safe alternative for both hybrids.

3.6 Discussion

In this paper, we have compared a wide array of recommenders and method configurations using two different input information sources: the ratings of users and their social context. Our analysis has focused on the tradeoff between the performance of each recommender method and its completeness, measured in terms of user coverage (ratio of users who receive a recommendation) and hitrate (ratio of users with at least one correct recommendation).

The first issue we have observed is that depending on the experimental configuration, the results could be biased towards one type of method or the other. If only users with a social context are tested (social evaluation, Section 3.2) social-based recommenders obtain very good results, and the same coverage as non-social methods. However, when considering users in the evaluation without social context (Section 3.3), the coverage of the social-based methods decreases, whereas the performance of non-social (collaborative) recommenders stays low. This situation motivates the definition of hybrid recommenders which make use of the two information sources (Section 3.4). We have experimented with different hybridization methods, namely feature combination and weighted hybrid recommenders according to Burke's taxonomy [BURKE, 2002]. Other hybridization methods could be explored, and are left for future work.

Performance of hybrid methods is usually close to (or even better than) that of the best recommender in the combination. Besides, these methods obtain coverage values as high as the best recommender being combined. Furthermore, by using indicators based on graph-based measures we have dynamically modified the weight given to each recommender in the combination on a user-basis. Results indicate that dynamic hybrid recommendation outperforms standard hybrids for different combinations of recommenders (Section 3.5).

A parameter we have found decisive in the tradeoff of performance and coverage is what we named overlap threshold. It determines the minimum number of neighbors required to have rated a particular item in order to consider a recommended score as a valid one. Results show that the larger this parameter is, the higher the performance for most of the recommenders, but the lower their coverage. This is because the recommenders have higher confidence in the actual suggestions, but have an additional constraint: it becomes more difficult to find interesting items for users with odd preferences.

Our study, thus, covers a range of variables, configurations and parameters, and their effect on the accuracy and coverage of different recommendation strategies. Further study

focusing on particular points addressed so far, e.g., sensitivity analyses, should be worthwhile as future work and might bring additional findings, by a more detailed observation of the dependency and variations in the investigated methods and strategies with respect to the relevant parameters.

4. RELATED WORK

Social Web systems deal with many different information sources, from user opinions about services and products, to items attributes and explicit ratings. The importance of such systems has grown in the last decade. In [KAUTZ et al., 1997], the authors propose to use different types of social networks (friends, colleagues, and collaborators) within an organization, in order to guide the search process, disambiguate queries and provide trusted recommendations by presenting named individuals when explaining suggestions. As in that work, social networks have been usually employed as a source of trust among individuals. Works like [O'DONOVAN & SMYTH, 2005] and more recently [JAMALI & ESTER, 2009], [MA et al., 2009], [WALTER et al., 2009], confirm this point. Other works, however, embed the social information into traditional CF algorithms in order to study how many neighbors automatically selected by the collaborative algorithm are socially connected [LEE & BRUSILOVSKY, 2010].

Information sources available in Social Web systems have, thus, been exploited from different points of view. Most proposed approaches, however, are only based on two dimensions of the data, mainly collaborative and social network information, such as [JAMALI & ESTER, 2010]. There exist, nonetheless, recent works dealing with further dimensions. In [BELLOGÍN et al., 2010], the authors make a comparative analysis of different types of recommenders, each of them making use of a different source of information: tags, ratings, and social links. [KONSTAS et al., 2009] propose a music recommender that combines tagging information, play counts, and social relations for making music track recommendations.

The lack of heterogeneous datasets probably explains, among the main reasons, the relatively small amount of works on Social Web systems that exploit more than the two aforementioned data dimensions. Nevertheless, recently in [JAMALI & ESTER, 2010], the authors make use of rating and social data, evaluating their recommenders with two manually crawled datasets: one from Epinions[4] (where a reputation system is used and various product categories are available) and another from Flixster[5] (the domain of this

[4] Epinions.com, General consumer review site, http://www.epinions.com
[5] Flixster Inc., Social movie site, http://flixster.com

dataset –movies– is the same as ours). Both datasets are publicly available, which, together with the Filmtipset dataset, as analyzed in this paper, may provide a basis for further research. In fact, the dataset used in this work may have a bigger impact, since it provides richer information such as the social user context, explicit ratings, temporal dimension, item reviews, and item attributes (directors/actors/scriptwriters).

It is important to note that as a consequence of combining different sources of information, recommendation evaluation measures have to be properly adapted, in order to consider every possible dimension in the recommendation task. As it was pointed out in [HERLOCKER et al., 2004], it is very important to go beyond accuracy metrics in order to define proper methodological targets for recommenders to achieve actual utility for real users. Coverage, novelty, and serendipity are some of the non-performance metrics mentioned by the authors as alternative relevant measures to be considered when building recommender systems. In this work, we have defined user coverage as the proportion of users able to receive a recommendation. Recently in [SHANI & GUNAWARDANA, 2011], the authors have defined what they called user space coverage, which is an analogous concept to what we define here. They propose an alternative way for computing coverage, which consists on measuring how rich a user's profile should be to receive recommendations, for example, by counting the number of items a user must rate before receiving recommendations. This recent research, among others, highlights the importance of this dimension when evaluating recommender systems, realizing the tradeoff between accuracy and coverage.

5. CONCLUSIONS

Aiming to explore how social context can be used to enhance recommendation, we have compared an array of different recommender systems under two evaluation scenarios: one in which only users with social relations are taken into account (social evaluation), and one where users with no friends are included in the test set (collaborative-social evaluation). The evaluated recommenders include collaborative filtering algorithms (such as user-based and similar users' popularity CF), social recommenders (such as friends' popularity, personal social, and pure social-based recommenders), and algorithms integrating collaborative and social information into hybrid recommendation methods (e.g., Random Walk-based recommenders). In order to uncover and compensate for the incompleteness of performance evaluations based on accuracy only, we have explored a new definition of a recently proposed metric to assess the user coverage of recommendation strategies.

The two considered evaluation scenarios, along with the new metric, have allowed us to analyze the different behavior of recommenders based on user ratings, social information, and an aggregation of both types of user information under different conditions. Our analysis confirms a shortcoming of social-based recommenders, namely their user coverage degrades when measured in a realistic scenario, that is, when the system includes users without friends. On the other hand, collaborative filtering algorithms show a higher user coverage (and lower performance). Therefore, we propose to use hybrid algorithms for combining the outputs from recommenders based on collaborative and social information. We show that recommenders using several information sources (such as those based on Random Walks and hybrids) obtain higher coverage than pure social-based recommenders.

Different techniques for combining recommenders based on collaborative and social information have been explored in the experiments. We first used linear (weighted) combinations of recommenders, observing mixed results: whereas the user coverage increased, the performance of the hybrid recommender was conditional to the original performance of each of the combined recommenders, in such a way that in some cases the hybrid recommender outperformed both baselines or at least the worst one. Secondly, we tested a novel approach to dynamically select the weight of each recommender on a user-basis, in such a manner that the social-based recommender is weighted more heavily on socially important users. Results were positive, outperforming the best static hybrid in different situations.

The generality of the observations deserves further study by testing them on other datasets, and heterogeneous ones in particular, where very different information is involved. Further analysis is required with respect to the collaborative-social evaluation. We plan to study to what extent the 50-50 proportion between users with and without social context is a reasonable representation of common social media systems currently in use. In that situation, the first evaluation performed in this paper would make sense and the results obtained would hold. Otherwise, if social media systems present different proportions, then this fact would encourage even further the use of hybrid recommenders. Further possibilities regarding hybrid recommendation can be explored, such as the extension of Markov chain-based recommenders (e.g., by using the techniques described in [FOUSS & PIROTE, 2007], which improve average computing times) and the analysis of different recommender combinations.

Apart from the *weighted* hybridization schema, other forms of hybrid recommenders (*switch*, *cascade*, and *meta-level*) should be investigated. Regarding the evaluated

personal social recommender [BEN-SHIMON et al., 2007], it is possible to investigate an alternative family of recommenders based on ratings instead of on graph distance. This new family would be able to suggest items even when social context is not available, since we may substitute the graph distance between users for a distance computed using a rating-based similarity measure.

ACKNOWLEDGMENTS

This work was supported by the Spanish Ministry of Science and Innovation (TIN2008-06566-C04-02), Universidad Autónoma de Madrid (CCG10-UAM/TIC-5877), and the Scientific Computing Institute at UAM. We gratefully acknowledge the comments of our reviewers. Their detailed comments and suggestions helped us to improve this article.

APPENDIX

A. THE ADJACENCY MATRIX $\mathcal{G}$ FOR RANDOM-WALK METHODS

Let us define a set of features $\mathcal{F} = \{F_1, \dots, F_P\}$, and consider an enumerated finite set $\mathcal{E} \subseteq \mathbb{N}$. Then, for all the elements in $\mathcal{F}$ we define

$$\begin{aligned} e\colon \mathcal{F} &\longrightarrow \wp(\mathcal{E}) \\ F_\cdot &\longrightarrow e(F_\cdot) = \mathcal{E}_{F_\cdot} \end{aligned} \tag{21}$$

being $\wp(\mathcal{E})$ the power set of the set $\mathcal{E}$. For example, when $F_1 \coloneqq$ ratings, then this set is $e(F_1) = \{1,2,3,4,5\}$. Moreover, for each feature $F_k \in \mathcal{F}$ we define

$$\begin{aligned} h_{F_k}\colon \mathcal{U} \times \mathcal{I}|_{F_k} &\longrightarrow E_{F_k} \\ (u,i) &\longrightarrow h_{F_k}(u,i) \end{aligned} \tag{22}$$

where h_{F_k} is the function that assigns to each pair of elements a value from the enumerated set related to the feature chosen, as defined in (21). The reader should take into account that the set $\mathcal{U} \times \mathcal{I}|_{F_k}$ is restricted to features applicable to (or relate) any pair of users and items in the collection, such as tags, and implicit/explicit preference.

Now, we define a user (respectively item) partition depending on whether there is a relation between a particular user (item) and the rest of items (users).

$$\forall u \in \mathcal{U}; \mathcal{I} = \{i \in \mathcal{I} / \exists h_{F_k}(u,i)\} \cup \{i \in \mathcal{I} / \nexists h_{F_k}(u,i)\} = \mathcal{I}^{+}_{F_{k,u}} \cup \mathcal{I}^{-}_{F_{k,u}} \tag{23}$$

$$\forall i \in I; \mathcal{U} = \{u \in U / \exists h_{F_k}(u,i)\} \cup \{u \in \mathcal{U} / \nexists h_{F_k}(u,i)\} = \mathcal{U}^{+}_{F_{k,i}} \cup \mathcal{U}^{-}_{F_{k,i}} \tag{24}$$

For instance, the set $\mathcal{U}^{+}_{F_{k,i}}$ represents the subset of users related to the item i with respect to the feature F_k. Once the subsets $\mathcal{U}^{+}_{F_{k,i}}$ and $\mathcal{U}^{-}_{F_{k,i}}$ have been defined, it is possible to obtain the average value of each feature F_k from two points of view: averaging over all users (equation 25) or over all items (equation 26).

$$\bar{h}_{F_k}(\cdot, i) = \frac{1}{\left| U^+_{F_{k,i}} \right|} \sum_{v \in \mathcal{U}^+_{F_{k,i}}} h_{F_k}(v, i) \tag{25}$$

$$\bar{h}_{F_k}(u, \cdot) = \frac{1}{\left| I^+_{F_{k,u}} \right|} \sum_{j \in \mathcal{I}^+_{F_{k,u}}} h_{F_k}(u, j) \tag{26}$$

Consider now the active user $u_m \in \mathcal{U}$. For each pair of users u_m, u, we define the following set

$$J^+_{F_{k,u,u_m}} = \mathcal{I}^+_{F_{k,u}} \cap \mathcal{I}^-_{F_{k,u_m}}, \tag{27}$$

which represents the items related to user u but not to the active user, with respect to the feature F_k.

Taking into account all the previous definitions, now we can define the adjacency matrix $\mathcal{G}$. Depending on the nature of each feature, we need to distinguish two different types of relations: explicit and implicit.

Explicit relations case

These types of relations are those that are directly extracted from the features. In this case the elements of $\mathcal{G}$ are defined as

$$\begin{aligned} g_{F_k}: \mathcal{U} \times \mathcal{I} &\longrightarrow E_{F_k} \\ (u, i) &\longrightarrow g_{F_k}(u, i) = h_{F_k}(u, i) \end{aligned} \tag{28}$$

For example, when $F_1 \coloneqq \text{rating}$ we have that

$$g_{\text{rating}}(u, i) = h_{\text{rating}}(u, i) = \begin{cases} 0 & u \text{ has not rated } i \\ rat(u, i) & \text{otherwise} \end{cases}$$

For this feature, we have that $\mathcal{I}^+_{F_{k,u_m}} = \mathcal{I}_m$.

Implicit relations case

In this case, we use a weighting vector $\vec{\alpha} \in \{0,1\}^{|\mathcal{F}|}$ to make use of various features. In this way, we are able to adjust the weight assigned to each available feature. Therefore, we define the elements of $\mathcal{G}$ as

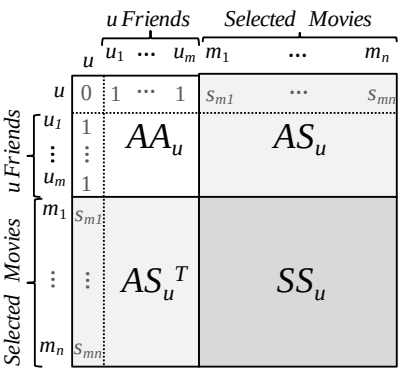
$$\begin{aligned} g_{F_k}: \mathcal{U} \times \mathcal{I} \times \{0,1\}^{|\mathcal{F}|} &\longrightarrow \mathbb{R} \\ (u, i, \vec{\alpha}) &\longrightarrow g_{F_k}(u, i, \vec{\alpha}) \end{aligned}$$

$$g_{F_k}(u, i, \vec{\alpha}) = \begin{cases} \bar{h}_{F_k}(\cdot, i) & i \in J^+_{F_{k,u,u_m}} \\ \dfrac{1}{\sum_{F_c \in \mathcal{F}} \arg max\left(E_{F_c}\right)} \displaystyle\sum_{F_c \in \mathcal{F}} \alpha_c \, \bar{h}_{F_c}(u, i) & i \notin J^+_{F_{k,u,u_m}} \end{cases} \tag{29}$$

Implicit relations take different values depending on whether the user is related to that particular item, where the assigned value is the average feature value for that item. Otherwise, the value is an average across the different available features, weighted by vector $\vec{\alpha}$.

In summary, the elements of the matrix $\mathcal{G}$ may be defined by expression (28) or (29), depending on the input data available. Moreover, the definition of function h_{F_k} may involve other input spaces different to user and items, such as users or items only. Examples of features defined in these spaces are friendship ($h_{\text{friendship}}(u,v) = 1$ if both users are friends) or similar item ($h_{\text{similar}}(i,j) = sim(i,j)$). Figure 7 shows which sub-matrices are needed, and how we can combine them in order to create the complete matrix $\mathcal{G}$.

Figure 7. Adjacency matrix $\mathcal{G}$ and their sub-matrices for the user u.



B. ANALYSIS OF THE TIE-BREAKING BIAS IN RECOMMENDER RANKINGS

The experimental results presented in this work are based on the evaluation of rankings generated by several recommenders. There is, however, an important concern when such an evaluation is performed, since the use of ranking cutoffs may introduce an uncontrolled parameter, or bias, depending on how the (possible) ties in the ranking –that is, two items with the same predicted score for some user– are broken. In this section, we measure how many ranking ties are generated for several recommender algorithms at different cutoffs. In particular, we analyze the number of items per user with ties at the different cutoffs used in this work –i.e., 5, 10, and 50– for the recommenders involved in our analysis. Furthermore, we compare our results with those reported in [CABANAC et al., 2010], where the authors compared many different TREC datasets with the *trec_eval* program, the same program we have used in our experiments. The authors observed that, in TREC, the tie-breaking strategy followed by this program is unfair. In the following, we check whether the results found for TREC appear in the recommender evaluation, and, in that case, if they are comparable with our results.

Table VI. Average ratio of tied items per user, at different cutoffs for the evaluated recommenders.

	Tied items at 5			Tied items at 10			Tied items at 50		
Recommender type	**Min**	**Avg**	**Max**	**Min**	**Avg**	**Max**	**Min**	**Avg**	**Max**
UB	15.11	130.64	280.83	14.33	130.52	280.83	7.83	128.20	281.39
SimPop	4.50	235.31	736.50	4.50	234.58	736.5	0	224.02	736.50
SVD	0	0	0	0	0	0	0	0.01	0.67
PureSocial	2	50.75	172	0	50.61	172	0	46.51	172
FriendsPop	10	350.20	1057	9	349.91	1052	0	343.13	1012
Personal	0	1.80	65	0	2.53	65	0	8.86	122.50
Combined	2.80	62.20	205.10	2	61.99	205.1	0.10	57.81	205.10

Table VI shows a typically unnoticed result about recommendation rankings. We observe that those methods, which directly depend on the size of some particular set, generate more ties in the ranking. This is the case of the *SimPop* recommender (where the items are ranked according to the number of users who have ranked each item), *FriendsPop* (like the previous one, but only considering the user's friends), and *PureSocial* (again, the number of friends of each user is very important). Several ties are also observed in the user-based recommender, this might be due to the fact that several neighbors obtain the same similarity value, along with the discrete scale of ratings, which makes very likely that different items obtains the same score.

On the other hand, recommenders like *SVD* and *Personal* generate very few tied items. This is probably because the formula used for estimating the preference is not normalized (*Personal*) or comes from more complicated equations (*SVD*). It is also noticeable that the hybrid recommender *Combined* provides a lower number of tied items with respect to one of the recommenders being combined, user-based CF.

In summary, the number of tied items per user in recommender systems is larger than expected and depends on the recommender under consideration. If we compare these results with those presented in [CABANAC et al., 2010] we find no direct equivalence between them, since the number of users (queries) and items (documents) is different. The conclusion, however, is the same for both: the issue of tie breaking affects the final results of the systems and, furthermore, it may even influence the final user satisfaction because of the so-called choice overload effect and item set attractiveness [BOLLEN et al., 2010].

REFERENCES

ADOMAVICIUS, G., AND TUZHILIN, A. 2005. Toward the next generation of recommender systems: a survey and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, 17(6), 734–749.

BAEZA-YATES, R., AND RIBEIRO-NETO, B. 1999. *Modern Information Retrieval*. Addison Wesley, 1st edition.

BALTRUNAS, L. AND AMATRIAIN, X. 2009. Towards time-dependant recommendation based on implicit feedback. In *Proceedings of the 3rd ACM conference on Recommender systems* (RecSys'09). ACM, New York, NY, USA, 423–424.

BARMAN, K. AND DABEER, O. 2010. Local popularity based collaborative filters. In *Proceedings of the International Symposium on Information Theory* (ISIT '10), 1668–1672.

BELLOGÍN, A., CANTADOR, I., CASTELLS, P. 2010. A study of heterogeneity in recommendations for a social music service. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems* (HetRec '10). ACM, New York, NY, USA, 1–8.

BEN-SHIMON, D., TSIKINOVSKY, A., ROKACH, L., MEISLES, A., SHANI, G., NAAMANI, L. 2007. Recommender system from personal social networks. In *Proceedings of the 5th Atlantic Web Intelligence Conference* (AWIC 2007), 47–55.

BOLLEN, D., KNIJNENBURG, B.P., WILLEMSEN, M.C., GRAUS, M. 2010. Understanding choice overload in recommender systems. In *Proceedings of the 4th ACM conference on Recommender systems* (RecSys'10). ACM, New York, NY, USA, 63–70.

BRIN, S., AND PAGE, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7), 107–117.

BURKE, R. 2002. Hybrid recommender systems: survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), 331–370.

CABANAC, G., HUBERT, G., BOUGHANEM, M., CHRISMENT, C. 2010 Tie-breaking bias: effect of an uncontrolled parameter on information retrieval evaluation. In *Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation* (CLEF 2010). Springer, LNCS 6360, 112–123.

CARTERETTE, B. 2011. System effectiveness, user models, and user utility: a conceptual framework for investigation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '11). ACM, New York, NY, USA, 903–912.

CELMA, O. 2008. Music recommendation and discovery in the long tail. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain.

DÍEZ, F., CHAVARRIAGA, J. E., CAMPOS, P. G., AND BELLOGÍN, A. 2010. Movie recommendations based in explicit and implicit features extracted from the Filmtipset dataset. In *Proceedings of the RecSys'10 Challenge on Context-aware Movie Recommendation* (CAMRa'10). ACM, New York, NY, USA, 45–52.

DIJKSTRA, E.W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271.

FERNÁNDEZ, M, VALLET, D., CASTELLS, P. 2006. Probabilistic Score Normalization for Rank Aggregation. In *Proceedings of the 28th European Conference on Information Retrieval* (ECIR '06), 553–556.

FOUSS, F., AND PIROTE, A. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19(3), 355–369.

FREEMAN, L. C. 1977. A set of measures of centrality based on betweenness. *Sociometry*, 40(1), 35–41.

HERLOCKER, J. L., KONSTAN, J. A., RIEDL, J. T. 2002. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5(4), 287–310.

HERLOCKER, J. L., KONSTAN, J. A., TERVEEN, L. G., RIEDL, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5–53.

JAMALI, M. AND ESTER, M. 2009. Using a trust network to improve top-n recommendation. In *Proceedings of the 3rd ACM conference on Recommender systems* (RecSys '09). ACM, New York, NY, USA, 181–188.

JAMALI, M. AND ESTER, M. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 4th ACM conference on Recommender systems* (RecSys '10). ACM, New York, NY, USA, 135–142.

JANSEN, B. J. AND SPINK, A. 2003. An Analysis of Web Documents Retrieved and Viewed. In *Proceedings of the 4th International Conference on Internet Computing* (IC '03). CSREA Press, 65–69.

JÄRVELIN, K., AND KEKÄLÄINEN, J. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4), 422–446.

KAUTZ, H., SELMAN, B., SHAH, M. 1997. Referral Web: combining social networks and collaborative filtering. *Communications of the ACM*, 40(3), 63–65.

KLEINBERG, J.M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5), 604–632.

KONSTAS, I., STATHOPOULOS, V., JOSE, J. M. 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). ACM, New York, NY, USA, 195–202.

KOREN, Y. and BELL, R. 2011. Advances in collaborative filtering. In *Recommender Systems Handbook*, F. RICCI, L. ROKACH, B. SHAPIRA, P.B. KANTOR, Eds. Springer US, Boston, MA, 145–186.

LEE, D. H. AND BRUSILOVSKY, P. 2010. Social networks and interest similarity: the case of CiteULike. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia* (HT '10). ACM, New York, NY, USA. 151–156.

LINDEN, G., SMITH, B., YORK, J. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76–80.

LIU, F., AND LEE, H.J. 2010. Use of social network information to enhance collaborative filtering performance. *Expert Systems with Applications*, 37(7), 4772–4778.

MA, H., KING, I., LYU, M. R. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '09). ACM, New York, NY, USA, 203–210.

MA, H., YANG, H., LYU, M. R., KING, I. 2008. SoRec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and Knowledge Management* (CIKM'08). ACM, New York, NY, 931–940.

O'DONOVAN, J. AND SMYTH, B. 2005. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces* (IUI '05). ACM, New York, NY, USA, 167–174.

SAID, A., BERKOVSKY, S., DE LUCA, E. W. 2010. Putting Things in Context: Challenge on Context-Aware Movie Recommendation. In *Proceedings of the RecSys'10 Challenge on Context-aware Movie Recommendation* (CAMRa'10). ACM, New York, NY, USA, 2–6.

SARWAR, B., KARYPIS, G., KONSTAN, J., RIEDL, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web* (WWW 10). ACM, New York, NY, USA, 285–295.

SHANI, G. and GUNAWARDANA, A. 2011. Evaluating recommendation systems. In *Recommender Systems Handbook*, F. RICCI, L. ROKACH, B. SHAPIRA, P.B. KANTOR, Eds. Springer US, Boston, MA, 257–297.

SHEPITSEN, A., GEMMELL, J., MOBASHER, B., BURKE, R. 2008. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2nd ACM Conference on Recommender Systems* (RecSys '08). ACM, New York, NY, USA, 259–266.

WALTER, F. E., BATTISTON, S., SCHWEITZER, F. 2009. Personalised and dynamic trust in social networks. In *Proceedings of the 3rd ACM conference on Recommender systems* (RecSys '09). ACM, New York, NY, USA, 197–204.

WANG, P., HAWK, W.B., AND TENOPIR, C. 2000. Users' interaction with World Wide Web resources: an exploratory study using a holistic approach. *Information Processing and Management* 36, 229–251.

WATTS, D. J. AND STROGATZ, S. 1998. Collective dynamics of 'small-world' networks. *Nature* 393 (6684), 440–442.

Received December 2010; revised March 2011; accepted October 2011.