A Study of Heterogeneity in Recommendations for a Social Music Service

Alejandro Bellogín, Iván Cantador, Pablo Castells

Departamento de Ingeniería Informática
Universidad Autónoma de Madrid
Campus de Cantoblanco
28049 Madrid, Spain

{alejandro.bellogin, ivan.cantador, pablo.castells}@uam.es

ABSTRACT

We present a preliminarily study on the influence of different sources of information in Web 2.0 systems on recommendation. Aiming to identify which are the sources of information (ratings, tags, social contacts, etc.) most valuable for recommendation, we evaluate a number of content-based, collaborative filtering and social recommenders on a heterogeneous dataset obtained from Last.fm. Moreover, aiming to investigate whether and how fusion of such information sources can benefit individual recommendation approaches, we propose various metrics to measure coverage, overlap, diversity and novelty between different sets of recommendations. The obtained results show that, in Last.fm, social tagging and explicit social networking information provide effective and heterogeneous item recommendations. Moreover, they give first insights on the feasibility of exploiting the above non performance recommendation characteristics by hybrid approaches.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation, Performance.

Keywords

Recommender systems, information heterogeneity, Web 2.0, folksonomy, collaborative tagging, implicit ratings, social contacts.

1. INTRODUCTION

Social systems (also called Web 2.0 systems) facilitate the creation of user generated content in various formats. Users post comments and reviews, rate and tag resources, upload and share multimedia contents, communicate online with social contacts, maintain personal bookmarks, and contribute to wiki-style knowledge bases, among others.

The vast amount and heterogeneity of the available contents in social media overwhelm human information processing capabilities and raise a wide range of research challenges in information management and retrieval. One of the most relevant of these challenges is the so called item recommendation, i.e. suggesting users products or services they might be interested in, by taking into account or predicting their tastes, interests or goals, and without requiring to perform explicit searches.

HetRec '10, September 26, 2010, Barcelona, Spain.

In recent years, recommendation approaches for Web 2.0 systems have been proposed. Most of these approaches use individual sources of information: ratings, tags, social contacts, etc. For example, tagging information have been exploited by graph-based algorithms [12], clustering strategies [20], [24], and content-based collaborative filtering [28], and explicit friend relations have been used to estimate or enhance user rating information [4], [10]. Other approaches are hybrid strategies that combine several sources of information to provide recommendations. In general, social data is combined with other types of data, such as item consuming history [15] content and demographic features [19], [23], or click-through information [22].

Some of the above works show empirical comparisons of performance results obtained with recommenders built on different types of input. However, as far as we know, there are no rigorous studies about the influence of each source of information on the provided recommendations.

Moreover, in general, aspects such as the coverage, diversity or novelty of item suggestions given by a recommender when using different sources of information have been barely taken into consideration yet in the literature. We claim that analysing and exploiting the above characteristics lets us to build more effective and adaptive hybrid recommendation approaches.

Motivated by the above facts, in this paper, we raise and address the following research questions:

- **RQ1.** Which sources of information available in Web 2.0 systems are more valuable for recommendation?

  To address this question, we study several performance metrics, such as precision and recall, for recommendation approaches that exploit different sources of information: ratings, tag, social contacts, etc.

- **RQ2.** Do recommendation approaches exploiting different sources of information in Web 2.0 systems really offer heterogeneous item suggestions, from which hybrid strategies could benefit?

  To address this question, we study several non-performance metrics that measure item recommendation characteristics, such as coverage, overlap, diversity and novelty, on the recommendation approaches studied in RQ1.

In order to carry out this study, we have implemented a set of content-based, collaborative filtering and social recommendation

approaches for Web 2.0 systems, and we have built a dataset with information of different kinds obtained from Last.fm[1]. By using these recommenders and dataset, we conduct a preliminary twofold study. First, we compare the performance of the recommenders with well known precision, recall and ranking based metrics. Second, we compare additional characteristics of the recommenders with a number of novel metrics that measure coverage, overlap, diversity and novelty of and between ranked lists of items.

The rest of the paper is organised as follows. Section 2 describes relevant works related to our study. Section 3 presents the evaluated content-based, collaborative filtering and social recommendation approaches. Section 4 explains the experimental setup of the study, describing the utilised dataset, the followed evaluation protocol, and the proposed performance and non-performance metrics. Section 5 discusses results obtained in the conducted experiment, and Section 6 depicts future research lines.

2. RELATED WORK

With the advent of Web 2.0, a variety of new recommendation approaches have been proposed in the literature. Most of these approaches are based on the exploitation of social tagging information and explicit friendship relations between users.

In social tagging systems, such as Delicious[2], Flickr[3] or Last.fm, users annotate/tag resources (Web pages, photos, music tracks, etc.) for the purpose of personal multimedia content management, browsing and search. Interestingly, these personalisation functionalities can be extended to collaborative recommendation functionalities when the whole set of annotations [user-tag-resource] (known as folksonomy) are taken into account. A user's preferences are described in terms of his tags and tagged resources. Based on such a profile model, similarities with other users can be found, and item recommendations can be produced. Hotho et al. [12] present FolkRank, a PageRank-like algorithm applied to the tripartite graph formed by nodes associated to users, tags and items of a folksonomy, and weighted edges related to co-occurrences between users and tags, items and tags, and users and items. Other approaches, like those proposed by Niwa et al. [20], and Shepitsen et al. [24] attempt to cluster the tag space, aiming to minimise information redundancy and contextualise item recommendations. Zanardi and Capra [28] investigate an alternative approach that provides item recommendations in a content-based collaborative filtering fashion. In this paper, we evaluate a number of tag-based recommendation approaches [6] that are adaptations of TF-IDF [2] and BM25 [25] Information Retrieval models, and are inspired on previous works on folksonomy-based personalised Web search presented by Noll and Meinel [21], and Xu et al. [27].

Apart from social tagging, other Web 2.0 systems provide social networking functionalities. In these systems, users explicitly state friendship[4] relations with other users. The use of this explicit social information has recently started to receive attention in the recommender systems field [9], and is currently an active open research direction. Thus, for instance, Ben-Shimon et al. [4] present a collaborative filtering strategy that estimates the rating of an item for a user based on the item ratings provided by the user's friends. He et al. [10], on the other hand, exploit the user's friends' ratings in a probabilistic recommendation model.

As suggested by Bonhard and Sasse [5], we believe that recommender systems can be enhanced by combining relevant information that can be drawn from social network analysis, such as explicit networks of trust, with the matching capabilities of content-based and collaborative filtering recommendation strategies. In this line, the final goal of our research is to investigate effective hybrid recommendation strategies that adaptively merge and exploit the heterogeneous information available in Web 2.0 systems.

Hybrid recommendation approaches that combine different sources of social information, especially social tags and contacts, have already been proposed. Konstas et al. [15] investigate the application of a Random Walk based algorithm on graphs where the user, tag and item spaces are intra- and inter-linked. Musial [19] studies recommendation methods enhanced with social features of the networks and their members. Sen et al. [22] present an empiric comparison of a large number of recommenders that estimate item ratings by exploiting user tags, ratings and click-through data. Finally, Seth and Zhang [23] propose a Bayesian model-based recommender that leverages content and social data.

Along with this research on hybrid social recommendation approaches, to our knowledge, there are no rigorous studies yet about how and to which degree each of the available sources of information in Web 2.0 systems is valuable for effective item recommendations. We address this issue here with a broad perspective, not restricting our empirical study to an evaluation of recommenders in terms of performance metrics such as precision and recall only, but also considering a further variety of metrics that aim to capture non-performance measures of recommendation usefulness, such as coverage, diversity, novelty and overlap of recommendations.

3. EVALUATED RECOMMENDERS

Adomavicius and Tuzhilin [1] formulate the recommendation problem as follows. Let $\mathcal{U} = \{u_1,\ldots,u_M\}$ be a set of users, and let $\mathcal{I} = \{i_1,\ldots,i_N\}$ be a set of items. Let $g:\mathcal{U}\times\mathcal{I}\rightarrow\mathcal{R}$, where $\mathcal{R}$ is a totally ordered set, be a utility function such that $g(u_m, i_n)$ measures the gain of usefulness of item i_n to user u_m. Then, for each user $u \in \mathcal{U}$, we want to choose items $i^{max,u} \in \mathcal{I}$, unknown to the user, which maximise the utility function g:

$$\forall u \in \mathcal{U}, \quad i^{max,u} = \arg\max_{i \in \mathcal{I}} g(u,i)$$

Depending on the exploited source of information, and the way in which the utility function g is estimated for different users, the following two main types of recommender systems are commonly distinguished: 1) content-based recommender systems, in which a user is recommended items similar to those he preferred in the past, and, 2) collaborative filtering systems, in which a user is recommended items that people with similar tastes and preferences liked in the past. We extend this classification by considering social recommender systems, i.e. systems in which a user is recommended items that (explicit) friends liked in the past, as a case related but significantly different to collaborative filtering.

With the above formulation, in the next subsections, we present the content-based, collaborative filtering and social recommenders for Web 2.0 systems used in the empirical study presented herein.

[1] Last.fm, Internet radio and music catalogue, http://www.last.fm

[2] Delicious, Social bookmarking, http://delicious.com

[3] Flickr, Photo sharing, http://www.flickr.com

[4] There are social networking sites that utilise other types of social relations, like e.g. fans, followers and professional colleagues

3.1 Content-based recommenders

Many Web 2.0 systems allow users to create or upload content (items), annotate it with freely chosen words (tags), and share it with other users. The whole set of tags constitutes an unstructured collaborative classification scheme that is commonly known as folksonomy. This implicit classification is then used to search for and discover items of interest.

More formally, a folksonomy $\mathcal{F}$ can be defined as a tuple $\mathcal{F} = \{\mathcal{T}, \mathcal{U}, \mathcal{I}, S\}$, where $\mathcal{T} = \{t_1, \dots, t_L\}$ is the set of tags that comprise the vocabulary expressed by the folksonomy, $\mathcal{U} = \{u_1, \dots, u_M\}$ and $\mathcal{I} = \{i_1, \dots, i_N\}$ are respectively the set of users and the set of items that annotate and are annotated with the tags of $\mathcal{T}$, and $S = \{(u_m, t_l, i_n)\} \in \mathcal{U} \times \mathcal{T} \times \mathcal{I}$ is the set of assignments (annotations) of each tag t_l to an item i_n by a user u_m.

In this section, we present a number of content-based (CB) recommendation approaches that exploit tagging information available in Web 2.0 systems. These approaches, evaluated in [6], are based on user and item profiles defined in terms of lists (vectors) of weighted tags, and compute similarities between such vectors to provide personal recommendations.

We define the profile of user u_m as a vector $\mathbf{u}_m = (u_{m,1}, \dots, u_{m,L})$, where $u_{m,l}$ is a weight (real number) that measures the "informativeness" of tag t_l to characterise contents annotated by u_m. Similarly, we define the profile of item i_n as a vector $\mathbf{i}_n = (i_{n,1}, \dots, i_{n,L})$, where $i_{n,l}$ is a weight that measures the relevance of tag t_l to describe i_n. There exist different schemes to weight the components of tag-based user and items profiles. Some of them are based on the information available in individual profiles, while others draw information from the whole folksonomy.

The simplest approach for assigning a weight to a particular tag in a user or item profile is by counting the number of times such tag has been used by the user or the number of times the tag has been used by the community to annotate the item. Thus, our first profile model for user u_m consists of a vector $\mathbf{u}_m = (u_{m,1}, \dots, u_{m,L})$, where

$$u_{m,l} = tf_{u_m}(t_l),$$

$tf_{u_m}(t_l)$ being the tag frequency, i.e. the number of times user u_m has annotated items with tag t_l.

Similarly, the profile of item i_n is defined as a vector $\mathbf{i}_n = (i_{n,1}, \dots, i_{n,L})$, where

$$i_{n,l} = tf_{i_n}(t_l),$$

$tf_{i_n}(t_l)$ being the number of times item i_n has been annotated with tag t_l.

In an information retrieval environment, common keywords that appear in many documents of a collection are not informative, and are generally not helpful to distinguish relevant documents for a given query. To take this into account, the TF-IDF weighting scheme is usually applied to the document profiles [2]. We adopt that principle, and adapt it to social tagging systems, proposing a second profile model, defined as:

$$u_{m,l} = tfiuf_{u_m}(t_l) = tf_{u_m}(t_l) \cdot iuf(t_l),$$

$$i_{n,l} = tfiif_{i_n}(t_l) = tf_{i_n}(t_l) \cdot iif(t_l)$$

As an alternative to TF-IDF, the Okapi BM25 weighting scheme follows a probabilistic approach to assign a document with a ranking score given a query [25]. We propose an adaptation of such model by assigning each tag with a score (weight) given a certain user or item. Our third profile model has the following expressions:

$$u_{m,l} = bm25_{u_m}(t_l) =$$

$$= \frac{u_{m,l} \cdot (k_1 + 1)}{u_{m,l} + k_1\left(1 - b + b \cdot {|u_m|}/{avg(|u_m|)}\right)} \cdot iuf(t_l),$$

$$i_{n,l} = bm25_{i_n}(t_l) =$$

$$= \frac{i_{n,l} \cdot (k_1 + 1)}{i_{n,l} + k_1\left(1 - b + b \cdot {|i_n|}/{avg(|i_n|)}\right)} \cdot iif(t_l)\,,$$

where b and k_1 are set to the standard values of 0.75 and 2, respectively.

3.1.1 TF-based recommender

To compute the preference of a user for an item, Noll and Meinel [21] propose a personalised similarity measure based on the user's tag frequencies:

$$g(u_m, i_n) = tf_u(u_m, i_n) = \frac{\sum_{l: i_{n,l} > 0} tf_{u_m}(t_l)}{\max_{u \in \mathcal{U}, t \in \mathcal{T}}(tf_u(t))}$$

The model utilises the user's usage of tags appearing in the item profile, but does not take into account their weights in such profile. We have introduced a slight variation in the above formula with respect to its original definition, namely a normalisation factor that scales the utility function to values in the range [0,1], without altering the user's item ranking.

3.1.2 BM25-based recommender

Analogously to the similarity based on tag frequencies described in Section 3.1.1, but using a BM25 weighting scheme, we propose a similarity function that only takes into account the weights of the user profile. This recommendation models is defined as follows:

$$g(u_m, i_n) = bm25_u(u_m, i_n) = \sum_{(l | i_{n,l} > 0)} bm25_{u_m}(t_l)$$

3.1.3 TF-IDF Cosine-based recommender

Xu et al. [27] use the cosine similarity measure to compute the similarity between user and item profiles. As profile component weighting scheme, they use TF-IDF[5]. Following our notation, their approach can be defined as follows:

$$g(u_m, i_n) = cos_{tf\text{-}idf}(u_m, i_n) =$$

$$= \frac{\sum_l tf_{u_m}(t_l) \cdot iuf(t_l) \cdot tf_{i_n}(t_l) \cdot iif(t_l)}{\sqrt{\sum_l \left(tf_{u_m}(t_l) \cdot iuf(t_l)\right)^2} \cdot \sqrt{\sum_l \left(tf_{i_n}(t_l) \cdot iif(t_l)\right)^2}}$$

3.1.4 BM25 Cosine-based recommender

Xu et al. [27] also investigate the cosine similarity measure with a BM25 weighting scheme. They use that model on personalised Web Search. We adapt and define it for social tagging as follows:

[5] Xu et al. do not specify if they take user-based or item-based inverse tag frequencies, or both. We chose to use both, since this configuration gave the best performance values.

$$g(u_m, i_n) = cos_{bm25}(u_m, i_n) =$$
$$= \frac{\sum_l \left(bm25_{u_m}(t_l) \cdot bm25_{i_n}(t_l)\right)}{\sqrt{\sum_l \left(bm25_{u_m}(t_l)\right)^2} \cdot \sqrt{\sum_l \left(bm25_{i_n}(t_l)\right)^2}}$$

3.2 Collaborative filtering recommenders

Collaborative filtering (CF) techniques match people with similar preferences, or items with similar choice patterns by users, in order to make recommendations. Unlike CB methods, CF systems aim to predict the utility of items for a particular user according to the items previously evaluated by other users.

In general, CF is based on explicit numeric ratings, that is, the real utility of an item for a particular user is represented by the rating given by that user to the item. There are systems, however, where no explicit ratings are available, but where user interests can be inferred from implicit feedback information. In order to provide item recommendations in such systems, two plausible options exist: use recommenders that directly exploit implicit data [8], [13], [26], or transform implicit data into explicit ratings to apply standard CF algorithms [4], [7], [17].

As mentioned before and explained in Section 4, we have conducted preliminary experiments with a dataset obtained from Last.fm. In this system, there are no explicit ratings, but user activity data logs in the form (user, item, freq), where item is a music track listened by user, and freq represents the number of times item was listened by user. Aiming to transform these tuples into numeric ratings, we follow the approach presented by Baltrunas and Amatriain [3], which is based on Celma's studies [7]. This approach consists of taking into account the number of times each user has listened to an artist (or track), in such a way that the artists (tracks) located in the 80-100% interquintile range of the user's listening distribution receive a rating of 5 (in a five point scale), the next interquintile range is mapped to a rating of 4, and so on.

In the following subsections, we briefly describe the CF algorithms evaluated in our experiments.

3.2.1 User-based CF recommender

User-based CF techniques compare the target user's choices with those of other users to identify a group of "similar-minded" people (usually called neighbours). Once this group has been identified, those items chosen or highly rated by the group are recommended to the target user. More specifically, the utility gain function $g(u_m, i_n)$ is estimated as follows:

$$g(u_m, i_n) = \mathrm{C} \sum_{v \in N[u_m,k]} sim(u_m, v) \times rat(v, i_n)$$

where C is a normalisation factor, $rat(v, i_n)$ is the rating given by user v to item i_n, and $N[u_m, k]$ denotes the set (with size k) of neighbours of u_m. Similarity between users can be calculated by using different metrics: Pearson and Spearman's correlations, cosine-based distance, among others [1]. In this work, we use Pearson's correlation, which is defined as:

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

where $\overline{rat}(u)$ is the average of the ratings provided by user u.

3.2.2 Item-based CF recommender

Like user-based approaches, item-based CF techniques recognise patterns. However, instead of identifying patterns of similarity between user choices, they recognise patterns of similarity between the items themselves. In general terms, item-based CF looks at each item on the target user's list of chosen/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 [1]. More formally, the utility gain function $g(u_m, i_n)$ is estimated as follows:

$$g(u_m, i_n) = \mathrm{C} \sum_{j \in I_m} sim(i_n, j) \times rat(u, j)$$

where I_m is the set of items rated by user u_m. In this work, we use Pearson's correlation to calculate item similarities.

3.3 Social recommenders

Inspired on the approach presented by Liu and Lee [18], we propose two simple recommenders that incorporate social information into the user-based CF model.

3.3.1 Social recommender

Our first social recommender utilises the same formula as the user-based CF technique, but replacing the set of nearest neighbours by the active user's (explicit) friends. That is:

$$N[u_m, k] = N[u_m] = \{v \in \mathcal{U}: v \text{ is friend of } u_m\}$$

3.3.2 Combined (social+CF) recommender

Our second social recommender also utilises the user-based CF formula, but is based on all the active user's friends, as well as his most similar nearest neighbours, combining them into a new neighbour set:

$$N[u_m, k] = \{v \in \mathcal{U}: v \text{ is friend of } u_m\} \cup \{v \in \mathcal{U}: sim(u_m, v) \geq \rho_m\}$$

where $\rho_m > 0$ is the minimum similarity to be satisfied between the active user and his most similar neighbours.

4. EXPERIMENTAL SETUP

4.1 Dataset

In order to evaluate the presented content-based, collaborative filtering and social recommendation models, we need a dataset rich in social tagging, item rating/consumption, and social networking information. Analysing representative Web 2.0 systems, we identify that Last.fm can satisfy our needs, and build the above heterogeneous dataset from such system.

Moreover, we build our dataset aiming to obtain a representative set of users, covering all music genres. Thus, we first identify the most popular tags related to the music genres in Last.fm. Then, we use the Last.fm API to get the top artists tagged with the previous tags. For each artist, we gather his/her fans along with their direct friends. Finally, we retrieve all tags and tagged tracks of the user profiles. Filtered out 1) those users without listened/tagged tracks and friend relations within the obtained social network, and 2) those tracks not listened and tagged by the remaining users, the final dataset contains 111 users, 18,921 tracks, 6,753 distinct tags, 22,134 tag assignments (~200 per user), and 1,149 friend relations (~10 per user).

4.2 Evaluation protocol

Figure 1 depicts the followed experimental methodology. We randomly split the set of tracks tagged and listened by the users in the database in two subsets. The first subset contains 80% of the

items for each user, and is used to build (train) the recommenders. The second subset contains the remaining 20% of the items, and is used to evaluate (test) the recommenders.

Specifically, regarding recommender building, CB approaches are built with the whole tag-based profiles of the training tracks, and with those parts of the users' tag-based profiles formed by tags annotating the training tracks; CF approaches are built with only those ratings associated to pairs [user-track] in the training set; and finally, social approaches are built with all friend relations available in the user profiles. Regarding recommender evaluation, CB approaches are evaluated with the tag-based profiles of the test tracks, while CF and social recommenders are evaluated on the test tracks set. In all evaluations, a 5-fold cross validation procedure is performed.

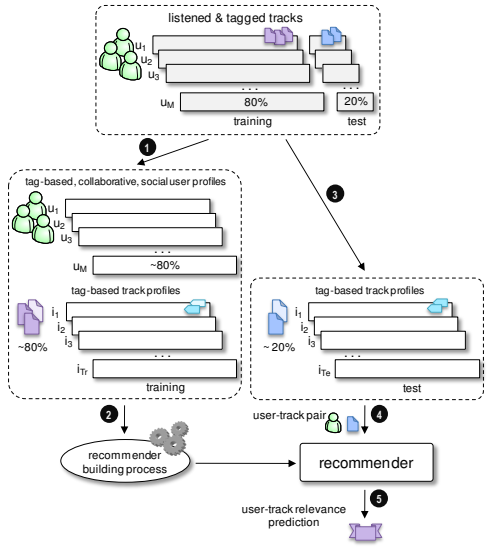

Figure 1. Description of the followed experimental methodology.

4.3 Performance metrics

In this section, we define the performance metrics used to empirically compare the implemented recommenders. Since in Last.fm, users do not explicitly rate items (tracks), metrics such as MAE (Mean Absolute Error) or RMSE (Root Mean Squared Error), commonly used in the recommender systems field, are not suitable to evaluate our recommendation algorithms.

For that reason, we shall measure the performance of the recommenders in terms of information retrieval metrics. We consider a content retrieval scenario where a system provides the user with a list of N recommended items. To evaluate the performance of each recommender, the selected metrics account for the ratio and position of relevant items in the ranked lists of recommended items. In our evaluation framework, the set of available items for recommendation is composed by all the items belonging to the test sets (see Section 4.2). As ground truth, we consider as relevant items for the active user those belonging to his test set, and all other items are considered non relevant.

4.3.1 Precision

Precision is defined as the ratio of recommended items that are relevant. If only the top N retrieved items are taken into consideration, the previous ratio is called Precision at N or P@N [2].

The average of P@N values at seen relevant items is called Mean Average Precision (MAP) [2]. MAP is a precision metric that emphasises ranking relevant documents higher.

Note that since in our experimental setting only the items in the user's profile are considered relevant, we cannot count potentially relevant items that the user has not seen, and we therefore get an underestimation of real precision (which is a known limitation of Information Retrieval metrics applied to recommender systems [11]). However, as the difference affects all the methods being evaluated, the metric is still consistent for comparative evaluations.

4.3.2 Recall

Recall is defined as the ratio of relevant items that are recommended. If only the top N recommended items are taken into consideration, the previous ratio is called Recall at N or R@N [2].

Again, it has to be noted that the considered set of relevant items is restricted to the items in the users' test sets, which is thus not complete (relevant items unknown to the users are not taken into account). We thus get an overestimation of recall, as we cannot evaluate whether the recommendation approaches are able to retrieve all relevant items but a representative sample of them.

4.3.3 Discounted Cumulative Gain

Precision and recall do not take into account the usefulness of an item based on its position in a result list. To address this issue, we also compute the Normalised Discounted Cumulative Gain (NDCG) metric [14].

NDCG penalises relevant items appearing lower in a result list. The penalisation is based on a relevance reduction logarithmically proportional to the position of the relevant items.

4.4 Non-performance metrics

In this section, we present a number of metrics to measure different non-performance characteristics of the recommenders. To better understand these metrics, in the following, we define several factors that appear in the metric formulations.

Let R_u be the set of items relevant for user u, and let $\mathcal{A}$ be the set of recommendation algorithms to be evaluated.

We define $L_{a,u}$, the ranked list of recommendations provided to user u by algorithm $a \in \mathcal{A}$, as:

$$L_{a,u} = \{(u,i,\tau): i \in \mathcal{I}, \tau > 0\},$$

where τ is the ranking position of item i in the recommendation list based on the predicted item utility $g_a(u,i)$, having $\tau_{a,u}(i) < \tau_{a,u}(j) \Rightarrow g_a(u,i) \geq g_a(u,j), \forall i,j \in \mathcal{I}$.

We denote by $S_{a,u}$ the set of items that belong to $L_{a,u}$:

$$S_{a,u} = \{i: (u,i,\cdot) \in L_{a,u}\}$$

Finally, we define $S_{a,u}^R$ as the set of those items belonging to $S_{a,u}$ that are relevant for user u. That is:

$$S_{a,u}^R = S_{a,u} \cap R_u = \{i: (u,i,\cdot) \in L_{a,u}, i \in R_u\}$$

The previous definitions $S_{a,u}$ and $S_{a,u}^R$ for a given recommendation algorithm a are extended to consider all users with the following expressions:

$$S_a = \bigcup_{u \in U} S_{a,u}, \quad S_a^R = \bigcup_{u \in U} S_{a,u}^R$$

Since some of the non-performance metrics explained below only depend on the top N recommendations provided by each algorithm a, we define $\overline{S}_{a,u}$, $\overline{S}_{a,u}^R$, $\overline{S}_a$ and $\overline{S}_a^R$ as, respectively, $S_{a,u}$, $S_{a,u}^R$, S_a and S_a^R on the set $L_{a,u}^N$ of top N recommendations for user u, where:

$$L_{a,u}^N = \{(\cdot,\cdot,\tau) \in L_{a,u}, \tau \leq N\}$$

4.4.1 Coverage

Coverage can be defined as the percentage of items for which a recommender $a \in \mathcal{A}$ can provide predictions [11]. Following the proposed notation, it is formulated as follows:

$$cvg(a) = \frac{|S_a|}{|I|}$$

Apart from this global coverage, we are also interested in measuring the percentage of relevant items a recommender is able to retrieve. For that purpose, we define coverage of relevant items as follows:

$$cvg^R(a) = \frac{|S_a^R|}{|\cup_{u \in U} R_u|}$$

4.4.2 Overlap

Aiming to measure the proportion of recommended items that are provided by two recommenders, we propose two overlap metrics for their recommended item lists. Both metrics are defined for the recommended items that are relevant for the users, and are limited to the top *N* results in each list.

Jaccard based overlap

The simplest approach to measure the overlap between two lists of items is by computing their intersection. Taking into account the cardinality of the sets of relevant items retrieved by the recommendation algorithms $a, b \in \mathcal{A}$, the intersection based overlap can be normalised by using the well known Jaccard similarity coefficient:

$$ove\_jacc(a,b) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} ove\_jacc_u(a,b)$$

$$ove\_jacc_u(a,b) = \frac{\left|\overline{S}_{a,u}^R \cap \overline{S}_{b,u}^R\right|}{\left|\overline{S}_{a,u}^R \cup \overline{S}_{b,u}^R\right|}$$

Ranking based overlap

The previous overlap metrics do not take into account the ranking position of relevant documents. Thus, for example, the lists of relevant items $L_{a,u}^R = \{i_1, i_2, i_3\}$ and $L_{b,u}^R = \{i_1, i_2, i_3\}$, would have the same overlap value than the lists $L_{a,u}^R = \{i_1, i_2, i_3\}$ and $L_{b,u}^R = \{i_3, i_1, i_2\}$, while the similarity between the given list is higher in the first case. As a rank-sensitive measure of overlap, we propose the following metric:

$$ove\_rank(a,b) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} ove\_rank_u(a,b)$$

$$ove\_rank_u(a,b) = \frac{1}{N} \sum_{i \in \overline{S}_{a,u}^R \cap \overline{S}_{b,u}^R} \left(1 - \frac{\left|\tau_{a,u}(i) - \tau_{b,u}(i)\right|}{N-1}\right)$$

As future work, we plan to test more sophisticated ranking overlap metrics, e.g. those proposed by Kumar and Vassilvitskii [16].

4.4.3 Diversity

A direct way to measure diversity is by computing entropy. In our context, the entropy based diversity for a recommender $a \in \mathcal{A}$ can be formulated as follows:

$$div(a) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} div_u(a)$$

$$div_u(a) = H_u(a) = -\sum_{i \in \overline{S}_{a,u}^R} p_{u,i} \cdot \log p_{u,i}$$

The open issue here is how to define the probability $p_{u,i}$ in terms of the diversity offered by item i for user u. Different approximations could be proposed. In this paper, we define $p_{u,i}$ in terms of item popularity among the evaluated recommenders. We assume that a recommender a provides diverse recommendations if these are not also recommended by a majority of the other recommenders for the same users. Formally, we set $p_{u,i}$ as follows:

$$p_{u,i} = \frac{\sum_{a \in \mathcal{A}} \delta(a,u,i)}{|\mathcal{A}|},$$

where $\delta(a,u,i) = 1$ iff $\in \overline{S}_{a,u}^R$, and 0 otherwise.

It is important to note that alternative definitions of diversity exist in the literature [1], and have to be investigated in the future.

4.4.4 Relative diversity

We can also measure diversity differences between two recommendation algorithms $a, b \in \mathcal{A}$ by computing their relative entropy:

$$div(a,b) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} div_u(a,b)$$

$$div_u(a,b) = H_u(a|b) = \sum_{i \in \overline{S}_{a,u}^R \cap \overline{S}_{b,u}^R} p_{a,u,i} \cdot \log \frac{p_{a,u,i}}{p_{b,u,i}}$$

Again, different approaches can be considered to define the probabilities $p_{a,u,i}$. In this case, given recommender a and user u, we assume a uniform distribution of items. That is:

$$p_{a,u,i} = \frac{1}{\left|\overline{S}_{a,u}^R\right|}$$

4.4.5 Novelty

Novelty can be defined in a twofold manner. On one hand, it can be defined as the capability of a recommender system to suggest a user with relevant items that have (usually content-based) characteristics not shared by items previously declared as relevant by the user. On the other hand, it can be defined in a more global way in terms of popularity among users [28], that is, as the capability of a recommender system to suggest a user with relevant but non popular items, i.e. items not liked or known by a wide number of users.

We follow here the second perspective and define novelty as follows:

$$nov(a) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} nov_u(a)$$

$$nov_u(a) = H_u(a) = -\sum_{i \in \overline{S}_{a,u}^R} p_i \cdot \log p_i\,,$$

where

$$p_i = \frac{|\{v \in \mathcal{U} : i \in R_v\}|}{|\mathcal{U}|}$$

This formula takes into account the proportion of users who are interested in each of the items retrieved by the recommender a that are relevant for user u.

5. RESULTS AND DISCUSSION

In this section, we present the performance and non performance values obtained with the proposed CF, CB and social recommenders. For those metrics that are defined in terms of the top N recommendations, we set $N = 20$. For CF and social approaches, we use 15 neighbours. We conducted experiments with other neighbourhood sizes, but we obtained worse results.

5.1 Recommendation performance

Table 1 shows the performance values of the recommenders. In general, CB approaches achieve better results than CF and social approaches. Analysing the characteristics of our dataset, we find out that its rating density ratio is 2.23%, while in other datasets such as MovieLens the rating density is around 4-6%. This may be the reason of the low results obtained with CF. As mentioned in Section 4.1, our dataset was built in such a way that all Last.fm music genres are covered by the evaluated tracks, which makes more difficult to get rating correlations between the few user profiles we have. Very interestingly, the social approach based on recommending a user items liked by explicit friends obtain acceptable precision and recall values. As concluded by Konstas et al. [15], in Last.fm, recommendations generated from the users' social networks represent a good alternative. Merging this approach with CF seems to not improve the results obtained with the approaches separately.

Table 1. Obtained performance values

	MAP	P@10	P@20	R@10	R@20	nDCG
cb-tf	0.010	0.298	0.170	0.014	0.014	0.170
cb-bm25	0.002	0.074	0.056	0.000	0.004	0.152
cb-cosine-tfidf	0.012	**0.316**	**0.244**	**0.016**	**0.022**	**0.220**
cb-cosine-bm25	**0.014**	0.244	0.196	0.010	**0.022**	0.212
cf-user	0.002	0.018	0.028	0.002	0.002	0.076
cf-item	0.000	0.000	0.010	0.000	0.000	0.068
social-friends	0.010	0.116	0.086	0.010	0.010	0.170
social-friends-cf	0.002	0.036	0.020	0.004	0.004	0.084

5.2 Recommendation coverage, diversity and novelty

Table 2 shows coverage, diversity and novelty values of the recommenders. In accordance with the obtained precision and recall results, CB approaches have higher coverage than CF and social approaches. CF and social approaches provide, however, more diverse and novel recommendations. These results are expected because of the well known over-specialisation limitation of CB techniques [1]. The social recommender based on explicit friends provides higher diversity and novelty than CF, fact that does not imply a loss of precision on the recommendations, as shown in Section 5.1. Moreover, these conclusions also give insights that the proposed metrics seem to really capture coverage, diversity and novelty characteristics of recommendation lists.

Table 2. Obtained coverage, diversity and novelty values

	cvg	cvg^R	div	nov
cb-tf	**0.017**	1.000	0.011	0.003
cb-bm25	**0.017**	1.000	0.008	0.001
cb-cosine-tfidf	**0.017**	1.000	0.018	0.004
cb-cosine-bm25	**0.017**	1.000	0.015	0.003
cf-user	0.015	1.000	0.005	0.001
cf-item	0.009	1.000	0.001	0.000
social-friends	0.013	1.000	**0.054**	**0.005**
social-friends-cf	0.013	1.000	0.004	0.001

Based on the conclusions given in Sections 5.1 and 5.2, we can provide a preliminary answer to **RQ1**, in the context of Last.fm. Social tags and explicit friends are sources of information that seem to provide accurate recommendations. With our dataset, implicit ratings from user listening logs give worse results.

5.3 Recommendation overlap

Tables 3 and 4 show overlap values between each pair of recommenders. Blank cells mean null overlap. Content-based recommenders overlap significantly between them. CF and friend-based social recommenders have some overlap with their combined approach. No overlap is found between content-based approaches and CF/social recommenders.

Although a more exhaustive study on the proposed overlap metrics has to be done, the obtained results allow us to give a preliminary answer to **RQ2**. The available sources of information in Web 2.0 systems can be exploited by hybrid recommenders to provide heterogeneous but valuable item recommendations. Such hybrid recommenders may merge or combine CB, CF and social recommendation approaches according to coverage, diversity and novelty characteristics of each approach for a particular user.

Table 3. Obtained Jaccard based overlap values

	cb-tf	cb-bm25	cb-cosine-tfidf	cb-cosine-bm25	cf-user	cf-item	social-friends	social-friends-cf
cb-tf		0.005	0.009	0.005				
cb-bm25			0.008	0.011	0.002			0.001
cb-cosine-tfidf				0.015				
cb-cosine-bm25								
cf-user								0.006
cf-item								
social-friends								0.003
social-friends-cf								

Table 4. Obtained ranking based overlap values

	cb-tf	cb-bm25	cb-cosine-tfidf	cb-cosine-bm25	cf-user	cf-item	social-friends	social-friends-cf
cb-tf		0.002	0.004	0.001				
cb-bm25			0.002	0.002				
cb-cosine-tfidf				0.005				
cb-cosine-bm25								
cf-user								0.002
cf-item								
social-friends								0.001
social-friends-cf								

5.4 Recommendation relative diversity

The proposed metric for relative diversity represents a first attempt to capture the information gain obtained with a recommender in comparison to other. For the evaluated recommenders, we obtain many null values. We think that this is due to the fact that the metric depends on the intersection of relevant items in the recommendation lists, which is, in general very low, as shown in Section 5.3. A further analysis on this issue has to be carried out in the future.

Table 5. Obtained relative diversity values

	cb-tf	cb-bm25	cb-cosine-tfidf	cb-cosine-bm25	cf-user	cf-item	social-friends	social-friends-cf
cb-tf								
cb-bm25	0.002			0.001				
cb-cosine-tfidf	-0.009			-0.016				
cb-cosine-bm25		-0.003	0.011					
cf-user								0.002
cf-item								
social-friends								
social-friends-cf					-0.003			

6. FUTURE WORK

We have presented a preliminary study on the influence of heterogeneous sources of information in Web 2.0 systems on recommendation. Content-based, collaborative filtering, and social recommenders have been empirically compared by using a variety of performance and non-performance metrics on a dataset obtained from Last.fm. We want to extend our investigation with more recommenders, hybridisation strategies, alternative mechanisms to transform implicit user preferences to explicit ratings, and additional datasets from other Web 2.0 systems such as Delicious and Flickr. Moreover, we are interested in considering the time dimension in recommendation, since such aspect was recognised as critical in the past NetFlix prize.

7. ACKNOWLEDGMENTS

This work was supported by the Spanish Ministry of Science and Innovation (TIN2008-06566-C04-02), and the Community of Madrid (S2009TIC-1542).

8. REFERENCES

[1] Adomavicius, G., 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.

[2] Baeza-Yates, R., Ribeiro-Neto, B. 1999. Modern Information Retrieval. Addison Wesley.

[3] Baltrunas, L., Amatriain, X. 2009. Towards Time-dependant Recommendation based on Implicit Feedback. In Proceedings of the RecSys 2009 Workshop on Context-aware Recommender Systems.

[4] Ben-Shimon, D., Tsikinovsky, A., Rokach, L., Meisles, A., Shani, G., Naamani, L. 2007. Recommender System from Personal Social Networks. In Proc. of the 5th Atlantic Web Intelligence Conference, 47-55.

[5] Bonhard P., Sasse M. A. 2006. Knowing Me, Knowing You - Using Profiles and Social Networking to Improve Recommender Systems. BT Technology Journal, 25(3), 84-98.

[6] Cantador, I., Bellogín, A., Vallet, D. 2010. Content-based Recommendation in Social Tagging Systems. In Proceedings of the 4th ACM Conference on Recommender Systems.

[7] Celma, O. 2008. Music Recommendation and Discovery in the Long Tail. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain.

[8] Das, A. S., Datar, M., Garg, A., Rajaram, S. 2007. Google News Personalization: Scalable Online Collaborative Filtering. In Proc. of the 16th International Conference on World Wide Web, 271-280.

[9] Guy, I., Chen, l., Zhou, M. X. (Eds.). 2010. IUI 2010 Workshop on Social Recommender Systems.

[10] He, J., Chu, W. W. 2010. A Social Network-Based Recommender System (SNRS). In Memon, N., Xu, J. J., Hicks, D. L., Chen, H. (Eds.), Data Mining for Social Network Data, 47-74.

[11] Herlocker, J. L, Konstan, J. A., Borchers, A., Riedl, J. 1999. An Algorithmic Framework for Performing Collaborative Filtering. In Proceedings of the 22nd Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval, 230-237.

[12] Hotho, A., Jäschke, R., Schmitz, C., Stumme, G. 2006. Information Retrieval in Folksonomies: Search and Ranking. In Proceedings of the 5th International Semantic Web Conference, 411-426.

[13] Hu, Y., Koren, Y., Volinsky, C. 2008. Collaborative Filtering for Implicit Feedback Datasets. In Proceedings of the 8th IEEE International Conference on Data Mining, 263-272.

[14] Jarvelin, K., Kekalainen, J. 2002. Cumulated Gain-based Evaluation of IR Techniques. ACM Transactions on Info. Systems, 20(4), 422-446.

[15] Konstas, I., Stathopoulos, V., Jose, J. M. 2009. On Social Networks and Collaborative Recommendation. In Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 195-202.

[16] Kumar, R., Vassilvitskii, S. 2010. Generalized Distances between Rankings. In Proceedings of the 19th International Conference on World Wide Web, 571-580.

[17] Lee, T., Park, Y., Park, Y. 2008. A Time-based Approach to Effective Recommender Systems using Implicit Feedback. Expert Systems with Applications, 34(4), 3055-3062.

[18] Liu, F., Lee, H. J. 2010. Use of Social Network Information to Enhance Collaborative Filtering Performance. Expert Systems with Applications, 37(7), 4772-4778.

[19] Musial, K. 2009. Recommender System for Online Social Network. Lambert Academic Publishing.

[20] Niwa, S., Doi, T., Honiden, S. 2006. Web Page Recommender *System based on Folksonomy Mining for ITNG'06 Submissions*. In Proceedings of the 3rd International Conference on Information Technology, 388-393.

[21] Noll, M. G., Meinel, C. 2007. Web Search Personalization via Social Bookmarking and Tagging. In Proceedings of the 6th International Semantic Web Conference, 367-380.

[22] Sen, S., Vig, J., Riedl, J. 2009. Tagommenders: Connecting Users to Items through Tags. In Proceedings of the 18th International Conference on World Wide Web, 671-680.

[23] Seth, A., Zhang, J. 2008. A Social Network Based Approach to Personalized Recommendation of Participatory Media Content. In Proceedings of the Intl. Conference on Weblogs and Social Media.

[24] 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, 259-266.

[25] Spärck-Jones, K., Walker, S., Robertson, S. E. 2000. A Probabilistic Model of Information Retrieval: Development and Comparative Experiments (parts 1 and 2). Information Processing and Management, 36(6):779-840.

[26] Wang, J., Robertson, S., de Vries, A., Reinders, M. 2008. Probabilistic Relevance Ranking for Collaborative Filtering. Information Retrieval, 11(6), 477-497.

[27] Xu, S., Bao, S., Fei, B., Su, Z., Yu, Y. 2008. Exploring Folksonomy for Personalized Search. In Proc. of the 31st Annual Intl. ACM SIGIR Conf. on Research and Development in Info. Retrieval, 155-162.

[28] Zanardi, V., Capra, L. 2008. Social Ranking: Uncovering Relevant Content using Tag-based Recommender Systems. In Proceedings of the 2nd ACM Conference on Recommender Systems, 51-58.