A New Approach to Evaluating Novel Recommendations

Òscar Celma
Music Technology Group
Universitat Pompeu Fabra
Barcelona, SPAIN
oscar.celma@iua.upf.edu

Perfecto Herrera
Music Technology Group
Universitat Pompeu Fabra
Barcelona, SPAIN
perfecto.herrera@iua.upf.edu

ABSTRACT

This paper presents two methods, named Item– and User–centric, to evaluate the quality of novel recommendations. The former method focuses on analysing the item–based recommendation network. The aim is to detect whether the network topology has any pathology that hinders novel recommendations. The latter, user–centric evaluation, aims at measuring users' perceived quality of novel recommendations.

The results of the experiments, done in the music recommendation context, show that *last.fm* social recommender, based on collaborative filtering, is prone to popularity bias. This has direct consequences on the topology of the item–based recommendation network. Pure audio content–based methods (CB) are not affected by popularity. However, a user–centric experiment done with 288 subjects shows that even though a social–based approach recommends less novel items than our CB, users' perceived quality is better than those recommended by a pure CB method.

Categories and Subject Descriptors

H3.3 [**Information Search and Retrieval**]: Information filtering, Selection process; G.2.2 [**Graph Theory**]: Graph algorithms

General Terms

Algorithms, Measurement, Human Factors

Keywords

recommender systems, evaluation, novelty, long tail, popularity, complex network analysis

1. INTRODUCTION

"If you like The Beatles you might like...$\mathcal{X}$". Now, ask several people and you will get lots of different $\mathcal{X}$'s. Each person, according to her ties with the band's music, would be able to propose interesting, surprising or expected $\mathcal{X}$'s. Nonetheless, asking the same question to different recommender systems one is likely to get similar results. Indeed, two out of five tested music recommenders contain John Lennon, Paul McCartney and George Harrison in their top–10 (*last.fm* and *the.echotron.com*). *Yahoo! Music* recommends John Lennon and Paul McCartney (1st and 4th position, respectively), whereas *Mystrands.com* only contains John Lennon (at top–10). Furthermore, Amazon's top–30 recommendations for the Beatles' *White Album* is strictly made of other Beatles' albums (all of a sudden, at the fourth page of the navigation there is the first non–Beatles album; *Exile on Main St.*, by The Rolling Stones)[1].

One can agree or disagree with all these lists of Beatles' similar artists. However, it is clear that there are a very few, if none at all, serendipitous recommendations (the rest of the similar artists were, in no particular order: The Who, The Rolling Stones, The Beach Boys, The Animals, and so on). Thus, novel recommendations are sometimes necessary in order to improve the user's experience and discovery in the recommendation workflow. The main goal of this paper is to evaluate the quality of novel recommendations in recommender systems, in terms of providing unknown, but relevant, items to users.

This paper is structured as follows: section 2 presents the background and related work to providing novel recommendations. Sections 3 and 4 present two new approaches, named respectively, Item– and User–centric evaluation, to evaluate different recommendation algorithms in terms of novelty. Then, section 5 presents the experiments performed in the context of the music domain, comparing three algorithms (collaborative filtering, content based, and a hybrid approach). Finally, the paper discusses our main findings in section 6, and concludes in section 7.

2. BACKGROUND

There is no clear recipe for providing *good* and *useful* recommendations to users. Still, there are at least three key elements that should be taken into account. These are: novelty, familiarity, and relevance [8]. According to Wordnet dictionary[2], **novel** (*adj.*) has two senses: "new–original and of a kind not seen before"; and "refreshing–pleasantly new or different". Likewise, **familiar** (*adj.*) is defined as "well known or easily recognised". Ideally, a user should be familiar with some of the recommended items, in order to improve confidence and trust in the system. Also, some items

RecSys'08, October 23–25, 2008, Lausanne, Switzerland.
Copyright 2008 ACM 978-1-60558-093-7/08/10 ...$5.00.

[1] All websites were accessed during May, 2008.

[2] `http://wordnet.princeton.edu`

should be unknown to the user (discovering *hidden* items in the catalog). A system should also give an explanation of why those—unknown—items were recommended, providing a higher confidence and transparency on these recommendations. The difficult job for a recommender is, then, to find the proper level of familiarity, novelty and relevance for *each* user.

2.1 Related work

It has been largely acknowledged that serendipity and novelty are relevant aspects in the recommendation workflow [13]. Indeed, there is some existing work that explicitly addresses these aspects. For instance, five measures to capture redundancy are presented in [20]. Using these measures the system can infer whether an item—that is considered relevant—contains any novel information to the user. In [19], the authors define novelty in terms of the user knowledge, and her degree of interest in a given item. Weng et. al propose, in [18], a way to improve the quality and novelty of the recommendations by means of a predefined taxonomy of topics, and hot topic detection using association rules. Other proposals include disregarding items if they are too similar to something the user has already seen [5], or simple metrics to measure novelty and serendipity based on the average popularity of the recommended items [21]. In fact, recommenders that appropriately discount popularity may increase total sales [9]. Generally speaking, the most popular items in the collection are the ones with higher probability that a given user will recognise, or be broadly familiar with. Likewise, one can assume that items with fewer interaction—rating, purchasing, previewing—within the community of users are more likely to be unknown [21]. In this sense, the Long Tail distribution—in terms of popularity—of the catalog [2] assists us in deciding how novel or familiar an item could be.

Even though these approaches focus on providing novel and serendipitous recommendations, there is not any framework that consistently evaluates the provided recommendations. Thus, there is a need in designing evaluation metrics to deal with the effectiveness of novel recommendations, not only measuring prediction accuracy, but taking into account other aspects such as usefulness and quality [10, 1]. Novelty metrics should look at how well a recommender system made a user aware of previously unknown items, as well as to what extent users accept the new recommendations [10]. We present two complementary methods to analyse and evaluate novel recommendations. On the one hand, an **item–centric** evaluation method analyses the item–based recommendation network. The aim is to detect whether the intrinsic topology of the network has any pathology that hinders novel recommendations. On the other hand, a **user–centric** evaluation aims at measuring the perceived quality of the recommendations.

3. ITEM–CENTRIC EVALUATION

In this section, we propose several metrics to analyse an item–based recommendation graph; being nodes the items, and the edges denoting the (weighted) similarity among the items. The metrics are derived from Complex Network and Social Network analysis, and are presented in section 3.1. Afterwards, novelty analysis based on item popularity is presented in section 3.2.

3.1 Complex networks metrics

3.1.1 Navigation

The **average shortest path** (or mean geodesic length) measures the distance between two vertices i and j. They are connected if one can go from i to j following the edges in the graph. The path from i to j may not be unique. The minimum path distance (or geodesic path) is the shortest path distance from i to j, d_{ij}. The average shortest path in a network of size N is:

$$\langle d \rangle = \frac{1}{\frac{1}{2}N(N+1)} \sum_{i \geq j} d_{ij} \tag{1}$$

In a random graph, the average path approximates to $\langle d_r \rangle \sim \frac{logN}{log\langle k \rangle}$, where $\langle k \rangle$ denotes the mean degree of all the nodes. The longest path in the network is called its **diameter** (D). In a recommender system, mean geodesic length and diameter inform us about the global navigation through the recommendation network.

The **strong giant component** (SGC) of a network is the set of vertices that are connected via one or more geodesics, and are disconnected from all other vertices. Typically, networks possess one large component that contains a majority of the vertices. It is measured as the % of nodes that includes the giant component. In a recommender system, SGC informs us about the catalog coverage, that is the total percentage of available items the recommender recommends to users [10].

3.1.2 Connectivity

The **degree distribution** is the number of vertices linked to a vertex, usually denoted k. The degree distribution p_k is the number of vertices with degree k:

$$p_k = \sum_{v \in V \,|\, \deg(v)=k} 1 \tag{2}$$

where v is a vertex, and $\deg(v)$ is its degree. More frequently, the *cumulative degree distribution* (the fraction of vertices having degree k or larger), is plotted:

$$P(k) = \sum_{k'=k}^{\infty} p_{k'} \tag{3}$$

A cumulative plot avoids fluctuations at the tail of the distribution and facilitates the evaluation of the power coefficient γ, in case the network follows a power law. In a directed graph, that is when a recommender algorithm only computes the top–n most similar items, $P(k_{in})$ and $P(k_{out})$, the cumulative incoming (outcoming) degree distribution, are more informative. Cumulative degree distribution detects whether a recommendation network has some nodes that act as hubs. That is, that they have a large amount of attached links. This can clearly affect the recommendations and navigability of the network.

Another metric used is the **degree correlation**. It is equal to the average nearest–neighbour degree, k^{nn}, as a function of k:

$$k^{\text{nn}}(k) = \sum_{k'=0}^{\infty} k' p(k'|k) \tag{4}$$

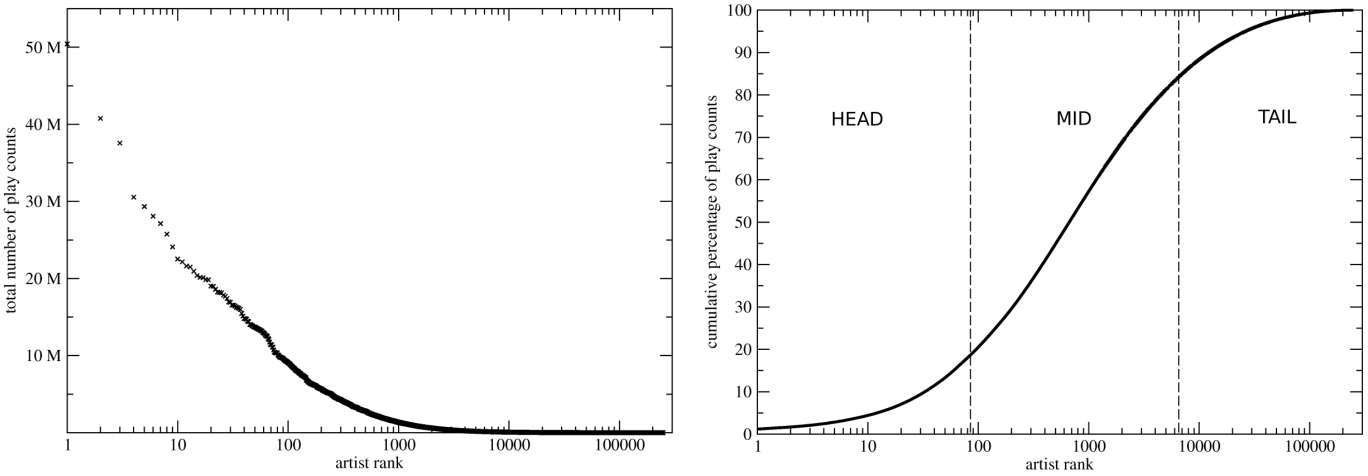

Figure 1: (left) The Long Tail of 260,525 music artists. A log–linear plot depicting artist rank in terms of total playcounts (e.g. at top–1 there is The Beatles with more than 50 million total playcounts). (right) Cumulative percentage of playcounts from the left figure. Top–737 artists accumulates the 50% of total playcounts (N_{50}). The curve is divided in three parts: head, mid and tail ($X_{head\to mid} = 82$, and $X_{mid\to tail} = 6,655$). The fitted model, $F(x)$, has $\alpha = 0.73$ and $\beta = 1.02$.

where $p(k'|k)$ is the fraction of edges that are attached to a vertex of degree k whose other ends are attached to vertex of degree k'. Thus $k^{\mathrm{nn}}(k)$ is the mean degree of the vertices we find by following a link emanating from a vertex of degree k.

A closely related concept is the **degree–degree correlation coefficient**, also named *assortative mixing*, which is the Pearson r correlation coefficient for degrees of vertices at either end of a link. In the case of a monotonically increasing (decreasing) k^{nn} means that high–degree vertices are connected to other high–degree (low–degree) vertices, resulting in a positive (negative) value of r [14]. In recommender systems, it measures to which extent nodes are connected preferentially to other nodes with similar characteristics.

3.1.3 Clustering

The clustering coefficient, C, estimates the probability that two neighbouring vertices of a given vertex are neighbours themselves. C is defined as the average over the *local measure*, C_i [17]:

$$C_i = \frac{2|E_i|}{k_i(k_i - 1)}, \tag{5}$$

where E_i is the set of existing edges that are direct neighbours of i, and k_i the degree of i. C_i denotes, then, the portion of actual edges of i from the potential number of total edges. For random graphs, the clustering coefficient is defined as $C_r \sim \langle k \rangle / N$. Typically, real networks have a higher clustering coefficient than C_r.

3.2 Novelty analysis based on item popularity

The previous section presented properties to analyse the topology of an item–based recommendation network. Now, we need to add the novelty component. The main idea is to correlate the above presented metrics with the Long Tail curve of the catalog. E.g. are the hubs in the network the most popular items? Are the most popular items connected with other popular items, and viceversa?

The Long Tail of a catalog is measured in terms of frequency distribution (e.g. purchases, downloads, etc.), ranked by popularity. Figure 1 (left) depicts the Long Tail for 260,525 music artists[3]. The horizontal axis contains the list of artists ranked by total playcounts. E.g. The Beatles, at position 1, have more than 50 million playcounts. We combine this information together with the recommendation network, to detect those items that could be both novel and relevant for a given user profile.

3.2.1 The Long Tail model

The Long Tail model, $F(x)$, simulates any heavy–tailed distribution [11]. It models the cumulative distribution (in %) of the Long Tail data. $F(x)$ equals to the share of total volume covered by objects up to rank x:

$$F(x) = \frac{\beta}{(\frac{N_{50}}{x})^{\alpha} + 1} \tag{6}$$

where α is the factor that defines the S–shape of the function, β is the total volume, and N_{50} is the number of objects that cover half of the total volume, that is $F(N_{50}) = 50$.

Once the Long Tail is modelled using $F(x)$, we can divide the curve in three parts: head, mid, and the tail part. The boundary between the head and the mid part of the curve is defined by:

$$X_{head\to mid} = N_{50}^{2/3} \tag{7}$$

Likewise, the boundary between the mid part and the end of the tail is:

$$X_{mid\to tail} = N_{50}^{4/3} \simeq X_{head\to mid}^2 \tag{8}$$

[3] The data was gathered from *last.fm* website during July, 2007. *Last.fm* provides plugins for virtually any desktop music player.

Figure 1 (right) depicts the cumulative distribution of the Long Tail of 260,525 music artists. Interestingly enough, the top–737 artists account for 50% of the total playcounts, $F(737) = 50$, and only the top–30 artists hold around 10% of the plays. In this sense, the *Gini coefficient* measures the inequality of a given distribution, and it determines the imbalance degree. In our Long Tail example, 14% of the artists hold 86% of total playcounts, yielding a Gini coefficient of 0.72. This value denotes an imbalanced distribution, higher than the 80/20 Pareto rule. Figure 1 (right) shows, too, the head of the curve, $X_{head \rightarrow mid}$ which consists of only 82 artists, whereas the mid part has 6573 ($X_{mid \rightarrow tail} = 6655$). The rest of the artists are located in the tail part.

3.2.2 Network analysis and the Long Tail model

Once each item is located in the head, mid, or tail part, the next step is to combine the properties of the items' similarity network with the Long Tail information. Two main analyses are performed. First, we measure item relationships in each part of the curve. That is, for each item that belongs to the head part, compute the percentage of similar items that are located in the head, mid and tail part (similarly, for the items in the mid and tail part). This measures whether the most popular items are connected to other popular items, and viceversa. Second, we measure the correlation between an item's location in the Long Tail and its indegree. This allows us to detect whether the hubs in the network are the most popular items. Section 5.1 presents the results comparing two different music artists recommendation algorithms: a social recommender based on collaborative filtering, and content–based audio filtering.

Item–centric evaluation measures the topology of the network, and combines this information with the Long Tail of the collection. Although, without any user intervention it is impossible to evaluate the quality and user satisfaction of the recommendations, which does not necessarily correlate with predicted accuracy [13]. The following section tries to overcome this limitation.

4. USER–CENTRIC EVALUATION

As of today, user–centric evaluation has been largely studied. The most common approaches are based on the *leave–n–out* method [6]. Given a dataset where a user has implicitly or explicitly interacted with (via ratings, purchases, downloads, previews, etc.), split the dataset in two disjunct sets: training and test. The evaluation of the accuracy is based only on a user's dataset, so the rest of the items of the catalog are ignored. The evaluation process includes several metrics such as: predictive accuracy (Mean Absolute Error, Root Mean Square Error), decision based (Mean Average Precision, Recall, F–measure, and ROC), and rank based metrics (Spearman's ρ, Kendall–τ, and half–life utility) [10].

The main problem, though, is developing evaluation metrics to deal with the effectiveness of the recommendations. That is, not only measuring prediction accuracy, but taking into account other aspects such as usefulness and quality [1].

4.1 Novelty analysis based on perceived quality

When evaluating serendipity and novelty, it is clear that feedback from the users is needed [13]. That is to say, users must examine the recommended items and measure, to some extent, whether they accept the list of recommendations. Figure 2 presents this evaluation method, we named **user–centric plus feedback**. The evaluation dataset is expanded to those items that the user has not yet seen (i.e. rated, purchased, previewed, etc.). The recommendation algorithm presents relevant items from outside the user's dataset, and asks for feedback. Feedback gathering can be done in two ways: implicitly or explicitly. Measuring implicit feedback includes, for instance, the time spent in an item's webpage, purchasing or previewing an item, etc. Explicit feedback is based on two related questions: whether the user already knew the item, and whether she likes it or not. Obviously, it requires an extra effort from the users, but at the same time it provides unequivocal information about the intended dimensions (which in the case of implicit measures could be ambiguous or inaccurate). Section 5.2 presents the results comparing three different music recommendation algorithms: collaborative filtering (CF), content–based audio filtering (CB), and a hybrid approach (HY).

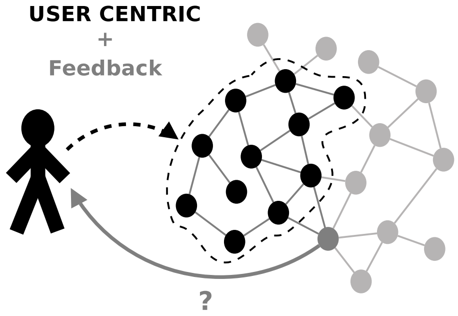

Figure 2: User–centric plus feedback evaluation method. The test dataset is expanded to those items that the user has not yet seen in the system. The system gets feedback of the recommendations in order to determine their quality.

5. EXPERIMENTS

In order to put into practice the proposed methods, we performed two experiments in the music recommendation field. It is worth noting that music is somewhat different from other entertainment domains, such as movies, or books. Tracking users' preferences are mostly done implicitly, via their listening habits. Moreover, a user can consume any song several times, even repeatedly and continuously. Regarding the evaluation process, music recommendation allows us instant feedback with a, say, 30 seconds excerpt.

The following section presents an item–centric evaluation comparing two different artists' networks, one based on collaborative filtering, and the other one based on content–based audio similarity. After that, section 5.2 presents the results for the recommended tracks, using the user–centric evaluation with explicit feedback.

5.1 Item–centric evaluation

The main goal here is to compare novel recommendations according to two different algorithms: a social recommender based on collaborative filtering (CF), and content–based audio filtering (CB). CF artist similarity was gathered from

Property	CF (*Last.fm*)	CB
N	122,801	59,583
$\langle k \rangle$	14.13	19.80
$\langle d_d \rangle$ ($\langle d_r \rangle$)	5.64 (4.42)	4.48 (4.30)
D	10	7
γ_{in}	2.31($\pm$0.22)	1.61($\pm$0.07)
r	0.92	0.14
C (C_r)	0.230 (0.0001)	0.025 (0.0002)

Table 1: Artist recommendation network properties for *last.fm* collaborative filtering (CF), and content–based audio filtering (CB). N is the number of nodes, and $\langle k \rangle$ the mean degree, $\langle d_d \rangle$ is the avg. shortest directed path, and $\langle d_r \rangle$ the equivalent for a random network of size N, and D is the diameter of the network. γ_{in} is the power–law exponent of the cumulative indegree distribution, and r is the indegree–indegree Pearson correlation coefficient (assortative mixing). C is the clustering coefficient, and C_r the equivalent for a random network.

last.fm, using Audioscrobbler webservices[4], and selecting the top–20 similar artists. *Last.fm* has a strong social component, and their recommendations are based on the classic item–based algorithm[5] [16]. To compute artist similarity, we use content–based audio analysis from a music collection ($\mathcal{T}$) of 1.3 Million tracks of 30 sec. samples. Audio analysis considers not only timbral features (e.g. Mel frequency cepstral coefficients), but some musical descriptors related to rhythm and tonality (e.g. key and mode) [7]. Then, to compute artist similarity we used the most representative tracks, $\mathcal{T}_a$, of an artist a, with a maximum of 100 tracks per artist. For each track, $t_i \in \mathcal{T}_a$, we obtain the most similar tracks (excluding those from artist a):

$$sim(t_i) = \underset{\forall t \in \mathcal{T}}{\operatorname{argmin}} \left(distance(t_i, t)\right), \qquad (9)$$

and get the artists' names, $\mathcal{A}_{sim(t_i)}$, of the similar tracks. The list of (top–20) similar artists of a is composed by all $\mathcal{A}_{sim(t_i)}$, ranked by frequency and weighted by the audio similarity distance:

$$similar\_artists(a) = \bigcup \mathcal{A}_{sim(t_i)}, \forall t_i \in \mathcal{T}_a \qquad (10)$$

5.1.1 Network analysis

Network properties of the two datasets are shown in Table 1. Both networks present the *small–world* phenomena [17]. They have a small average directed shortest path, $\langle d_d \rangle$, close to its equivalent random network, $\langle d_r \rangle$. Also the clustering coefficients, C, are significantly higher than C_r. This is an important property, because allows users surfing to any part of a music collection with a small number of mouse clicks, using only local information from the network [12].

The main differences between the two networks are the assortative mixing (Pearson r coefficient), and the power–law γ exponent. CF presents a high assortative mixing ($r = 0.92$). That means that the most connected artists are prone to be similar to other top connected artists. CB does not present any indegree correlation coefficient, thus artists are connected independently of their inherent properties. Figure 3 depicts this phenomena.

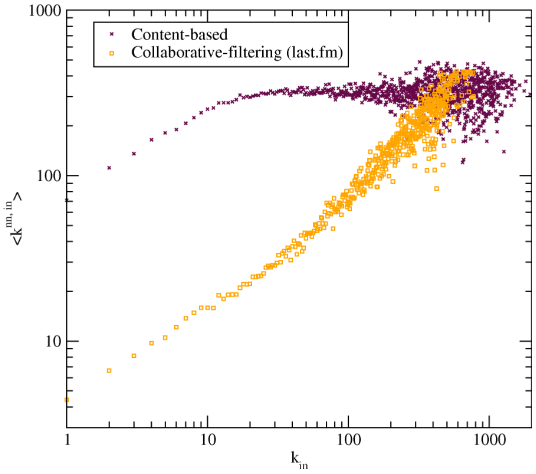

Figure 3: Item–centric evaluation. Assortative mixing (indegree–indegree correlation coefficient) for *last.fm* collaborative filtering (CF), and content–based (CB). CF presents assortative mixing ($r_{CF} = 0.92$), whereas CB does not ($r_{CB} = 0.04$).

Regarding the power–law γ exponent, CF has $\gamma = 2.31$, similar to those detected in many scale–free networks, including the world wide web linking structure [4]. These networks are known to show a right–skewed power law distribution, $P(k) \propto k^{-\gamma}$ with $2 < \gamma < 3$, relying on a small subset of hubs that control the network [3].

5.1.2 Novelty analysis based on artist popularity

Item–centric evaluation shows that the topology of the two networks is rather different. Now, we need to combine network analysis with the artists' Long Tail location. Table 2 presents artist similarity divided into the three sections of the curve (head, mid, and tail). Given an artist, a_i, it shows (in %) the Long Tail location of its similar artists (results are averaged over all artists). In the CF network, given a very popular artist from the head part, the probability of reaching (in one click) a similar artist in the tail is zero. Actually, half of the similar artists are located in the head part (82 artists), and the rest in the mid area. Artists in the mid part are tightly related to each other, and only 1/5 of the similar artists are in the tail part. Finally, given an artist in the tail, its similar artists remain in the same area. On the other hand, CB promotes much more the mid and tail parts in all the cases.

Another experiment analyses whether the hubs in the network (artists with higher indegree) are also the most popular artists. Figure 4 presents the results, and CF confirms the hypothesis; the artists with higher indegree are the ones with more playcounts, with a Pearson correlation value of $r_{CF} = 0.38$. In CB, hubs are more spread out through all the curve. Moreover, a Markovian stochastic process is used to simulate someone surfing the recommendation network. Indeed, each row in Table 2 can be seen as a Markov chain transition matrix, M, being the head,

[4] `http://www.audioscrobbler.net/data/webservices/`

[5] Although, it is clear that *last.fm* is using, as well, information gathered from social tagging.

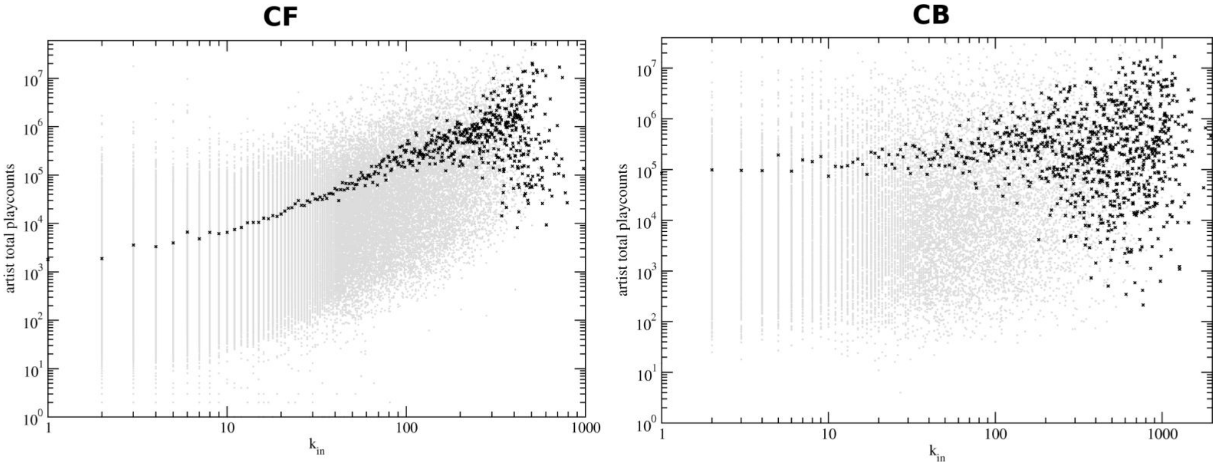

Figure 4: Item–centric evaluation. A log–log plot showing the correlation between artist indegree (k_{in}, in horizontal axis) and its total playcounts (avg. values in black), in vertical axis. Pearson r values are: $r_{CF} = 0.38$, $r_{CB} = 0.10$.

Method	$a_i \to a_j$	Head	Mid	Tail
CF top–20	**Head**	45.32%	54.68%	0%
	Mid	5.43%	71.75%	22.82%
	Tail	0.24%	17.16%	82.60%
CB top–20	**Head**	6.46%	64.74%	28.80%
	Mid	4.16%	59.60%	36.24%
	Tail	2.83%	47.80%	49.37%

Table 2: Item–centric evaluation. The table shows the similarities among artists, and their location in the Long Tail. Each row represents, also, the Markov chain transition matrix for CF and CB.

Method	k	$\mathbf{P^{(k)}}$
CF	5	$(0.075_H, 0.512_M, 0.413_T)$
CB	2	$(0.038_H, 0.562_M, 0.400_T)$

Table 3: Item–centric evaluation. Long Tail navigation in terms of a Markovian stochastic process. Second column depicts the number of clicks (k) to reach the tail from the head part, with a probability $p_{head,tail} \geq 0.4$. Third column shows the probability distribution after k clicks.

mid and tail parts the different states. The values of M denote the transition probabilities, $p_{i,j}$, between two states i, and j (e.g. $p^{CF}_{head,mid} = 0.5468$). The Markovian transition matrix, M^k, denotes the probability of going from any state to another state in k steps (clicks). The initial distribution vector, $P^{(0)}$, sets the probabilities of being at a determined state at the beginning of the process. Then, $P^{(k)} = P^{(0)} \times M^k$, denotes the probability distribution after k clicks, starting in the state defined by $P^{(0)}$. Using $P^{(k)}$ and defining $P^{(0)} = (1_H, 0_M, 0_T)$, we can get the probability of reaching the tail, starting in the head part. Table 3 shows the number of clicks needed to reach the tail from the head, with a probability $p_{head,tail} \geq 0.4$. In CF, one needs five clicks to reach the tail, whereas in CB only two clicks are needed.

Yet, we need to evaluate the quality of the relationships among artists, as well as the effects of the hubs when providing novel recommendations to the users. The following section is devoted to giving some insights to these questions.

5.2 User–centric evaluation

A user–centric evaluation, with explicit feedback, was performed in order to analyse three music recommendation algorithms (CF, CB and hybrid) when providing novel songs[6]. CF song similarity data come, again, from *last.fm*. For the CB method, audio similarity is based on equation 9. Hybrid method (HY) is based on combining related artists from *Allmusic.com* musicologists, and CB audio similarity at track level. Given a seed track, the most similar tracks are computed this way: first, select the related artists (according to the experts) from the artist seed track. Then, rank all the tracks from the related artists, according to the audio similarity.

5.2.1 Procedure

The experiment was based on providing *personalised* song recommendations to users, using some seed tracks from their top–20 most played artists, based on their *last.fm* profiles. Recommended songs (evenly distributed from CF, CB and HY) had no metadata displayed (neither artist name nor song title), but only a preview of 30 seconds. Provided feedback included whether the user knew the song (*no*, recall *only the artist*, recall *artist name and song title*), and the quality of the recommendation—whether she likes the song or not—on a rating scale from `1` (*I don't like it*) to `5` (*I like it very much*). After running the experiment during March 2008, 5,573 tracks were rated by 288 users (average of 19 tracks rated per user).

[6] The experiment is available at: `http://foafing-the-music.iua.upf.edu/survey`

Method	Case	%	Avg.Rating (Stdev)
CF	*Recall A&S*	15.50	4.64(±0.67)
	Recall only A	12.81	3.88(±0.99)
	Unknown	71.69	3.03(±1.19)
HY	*Recall A&S*	10.71	4.55(±0.81)
	Recall only A	10.95	3.67(±1.18)
	Unknown	78.34	2.77(±1.20)
CB	*Recall A&S*	10.50	4.56(±1.21)
	Recall only A	8.53	3.61(±1.10)
	Unknown	80.97	2.57(±1.19)

Table 4: User–centric evaluation. Novelty analysis for *last.fm* collaborative filtering (CF), Hybrid (HY), and audio content–based (CB) algorithms. *Recall A&S* means that a user recognises both *A*rtist name and *S*ong title.

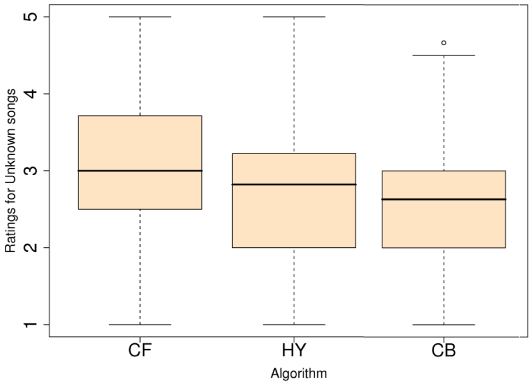

Figure 5: User–centric evaluation. Box–and–whisker plot showing the ratings for unknown songs.

5.2.2 *Novelty analysis based on perceived quality*

Table 4 presents the overall results for the three algorithms. It shows, for each algorithm, the percentage of songs that users identify in the recommendations (i.e. they are familiar with), as well as the novel, unknown ones, and the quality of the recommendations (average rating and standard deviation).

To compare the three algorithms we performed a one–way ANOVA within subjects. For familiar recommendations (including both *artist and song known*, and *recall only artist*) there is no statistically significant difference among the ratings of the three algorithms. The main differences are found in the ratings of unknown songs (3.03_{CF} vs. 2.77_{HY} vs. 2.57_{CB}, $F(2, 287) = 17.27$, with $p \ll 0.01$), and in the percentage of unknown songs ($71.69\%_{CF}$ vs. $78.34\%_{HY}$ vs. $80.97\%_{CB}$, $F(2, 287) = 32.69$, with $p \ll 0.01$). In the former case, Tukey's test for pairwise comparisons confirms that CF scores higher than HY and CB, at 95% family-wise confidence level. However, according to the latter case, CF generates more familiar songs than CB and HY (also validated by the corresponding Tukey's test). Thus, CB and HY provide more novel recommendations, although their quality is not as good as CF (see Figure 5).

6. DISCUSSION

The results from the item–centric analysis show that, using *last.fm* CF algorithm, the popularity effect that arose from the community has consequences in the recommendation network. This reveals a somewhat poor discovery ratio when just browsing through the network of similar music artists. It is not easy to reach relevant Long Tail artists, starting from the head or mid parts. Moreover, similar artists all located in the tail area do not always guarantee novelty. A user that knows quite well an artist in the Long Tail is likely to know most of the similar artists too (e.g. the solo project of the band's singer, collaborations with other musicians, and so on). Thus, these might not be considered good novel recommendations to that user, but familiar ones.

The key Long Tail area in CB are the artists located in the mid part. These artists allow users to navigate inside the Long Tail acting as entry points, as well as main destinations when leaving the Long Tail. Users that listen to mainly very unknown (Long Tail) music are likely to discover artists that are in the mid part, and that are easily reachable from the artist in the tail. One should pay attention, too, to the quality data in the Long Tail. Assuming that there exists some extremely poor quality items, CB is not able to clearly discriminate it. In some sense, the popularity effect drastically filters these low quality items. Although, as shown in [15] popularity is only partly determined by quality.

Regarding user–centric evaluation, in general, user perceived quality for novel, unknown, recommendations is on the negative side (avg. rating around 3/5 or less, see Table 4). This probably emphasises the need for adding more context when recommending unknown music. Users might want to understand why a song was recommended. Recommender systems should give as many reasons as possible, even including links to external sources (reviews, blog entries, etc.). Besides, the limitation in the experiment of using only 30 sec. samples did not help to assess the quality of the song. Yet, there are lots of industrial music recommender systems that can only preview songs due to licensing constraints. So, our experimental constraint is not that far from the reality.

An interesting result obtained is that, to provide familiar items, any of the three proposed algorithms works fine. This has some implications when designing a recommender system. For instance, to provide a *Radio–a–la–carte* experience, the system does not need to rely exclusively on millions of users, but it can do that quite well with a state–of-the–art audio similarity algorithm.

The context–free and popularity agnostic CB algorithm sometimes points in the wrong direction (it is not that easy to discriminate between a, say, classical guitar and a harpsichord, based solely on the audio content), and gives poor or non–sense recommendations. This leaves room for improvement the audio similarity algorithm. In this sense, the proposed Hybrid approach drastically reduces the space of possible similar tracks, to those artists related with the original artist. This avoids, most of the time, the *mistakes* performed by the pure CB. CF tends to be more conservative, providing less novel recommendations, but of higher quality.

We can envision different solutions to cope with novelty in recommender systems. The first one is to use CF, promoting unknown artists by means of exploiting the Long Tail information of the catalog, and the topology of the recommendation network. Another option is switching among algorithms when needed. E.g. to avoid the cold–start problem and, at the same time, to promote novelty, the best option is to use CB or HY. After a while, the system can move to a stable CF or HY approaches. Moreover, the system should be able to change the approach according to the user's needs. Sometimes, a user is open to discovering new artists and songs, while sometimes she just wants to listen to her favourites. Detecting these modes and acting accordingly would increase user's satisfaction with the system. However, this leads us to the topic of user profiling, which is a different kind of problem than the one dealt in this paper.

7. CONCLUSIONS

In this paper we have presented two methods, named Item– and User–centric evaluation. The former focuses on analysing the topology of the recommendation network, and then combining the results with the popularity of the items. The latter method, user–centric aims at measuring the perceived quality of the recommendations using explicit feedback.

The results of the experiments show that CF is prone to popularity bias, affecting both the topology of the network, and the novelty recommendation ratio, whilst pure CB is not affected by popularity. However, a user–centric experiment performed with 288 subjects and 5,573 tracks shows that, even though CF recommends less novel items than CB, users' perceived quality is higher than that for those recommended by pure CB.

Future work includes expanding the analysis of the recommendation network, taking into account its dynamics. This could be used, for instance, to detect "hype" items. Regarding user–based recommendation algorithms, a similar network analysis can be applied, now the nodes being users. An interesting feature would be to detect trendsetters, and its effect when providing novel recommendations.

8. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] C. Anderson. *The long tail. Why the Future of Business is Selling Less of More.* Hyperion Verlag, 2006.

[3] A. L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.

[4] A.-L. Barabási, R. Albert, H. Jeong, and G. Bianconi. Power-law distribution of the world wide web. *Science*, 287:2115a, 2000.

[5] D. Billsus and M. J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3):147–180, 2000.

[6] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. Technical report, 1998.

[7] P. Cano, M. Koppenberger, and N. Wack. An industrial-strength content-based music recommendation system. In *Proceedings of 28th International ACM SIGIR Conference*, Salvador, Brazil, 2005.

[8] Ò. Celma and P. Lamere. Music recommendation tutorial. In *Proceedings of 8th International Conference on Music Information Retrieval*, Vienna, Austria, 2007.

[9] D. M. Fleder and K. Hosanagar. Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity. *SSRN eLibrary*, 2007.

[10] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[11] K. Kilkki. A practical model for analyzing long tails. *First Monday*, 12(5), May 2007.

[12] J. M. Kleinberg. Navigation in a small world. *Nature*, 406:845, 2000.

[13] S. M. Mcnee, J. Riedl, and J. A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *Computer Human Interaction. Human factors in computing systems*, pages 1097–1101, New York, NY, USA, 2006. ACM.

[14] M. E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20), 2002.

[15] M. J. Salganik, P. S. Dodds, and D. J. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311(5762):854–856, February 2006.

[16] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In A. Press, editor, *Proceedings of 10th International World Wide Web Conference*, pages 285–295, Hong Kong, 2001.

[17] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[18] L.-T. Weng, Y. Xu, Y. Li, and R. Nayak. Improving recommendation novelty based on topic taxonomy. In *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pages 115–118, Washington, DC, USA, 2007. IEEE Computer Society.

[19] Y. Yang and J. Z. Li. Interest-based recommendation in digital library. *Journal of Computer Science*, 1(1):40–46, 2005.

[20] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th international ACM SIGIR conference*, pages 81–88, New York, NY, USA, 2002. ACM.

[21] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32, New York, NY, USA, 2005. ACM.