Combining Usage and Content in an Online Music Recommendation System for Music in the Long-Tail

Marcos A. Domingues
INESC TEC, Portugal
marcos.a.domingues@inescporto.pt

Fabien Gouyon
INESC TEC, Portugal
fgouyon@inescporto.pt

Alípio Mário Jorge
FCUP, University of Porto
INESC TEC, Portugal
amjorge@fc.up.pt

José Paulo Leal
CRACS, INESC TEC, FCUP
University of Porto, Portugal
zp@dcc.fc.up.pt

João Vinagre
FCUP, University of Porto
INESC TEC, Portugal
jnsilva@inescporto.pt

Luís Lemos
University of Porto
INESC TEC, Portugal
llemos@inescporto.pt

ABSTRACT

In this paper we propose a hybrid music recommender system, which combines usage and content data. We describe an online evaluation experiment performed in real time on a commercial music web site, specialised in content from the very long tail of music content. We compare it against two stand-alone recommenders, the first system based on usage and the second one based on content data. The results show that the proposed hybrid recommender shows advantages with respect to usage- and content-based systems, namely, higher user absolute acceptance rate, higher user activity rate and higher user loyalty.

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: Artificial Intelligence; I.2.6 [**Artificial Intelligence**]: Learning—*Induction*

General Terms

Algorithms, Experimentation, Measurement

Keywords

Music recommendation, hybrid recommender system, usage data, tags, audio features

1. INTRODUCTION

Music discovery and consumption has changed dramatically in recent years. According to recent reports, e.g. from consultancy firms [6], the web has become an increasingly relevant source of music discovery, recently reaching the importance of traditional sources such as AM/FM radios, music TVs, or friends. Most people now consume music on their personal computers and mobile devices via Internet. However, with virtually millions of pieces of music –henceforth tracks– available from thousands of web sites or online services, avoiding overwhelming choices and finding the "right" music has become a challenge for users. Music recommender systems have emerged in response to this problem. A music recommender system is an information filtering technology which can be used to output an ordered list of music tracks that are likely to be of interest to the user [2].

Music recommendation has flourished on the Internet, and web sites as Last.fm[1], Amazon[2] and Pandora[3] are successful examples of music recommenders that adapt recommendations to particular user tastes. In this paper we present a hybrid recommender system implemented for Palco Principal[4], a web site of Portuguese music. Most of its music tracks are underground, unknown/unpopular and rarely accessed/rated by the users. In fact, only 19.7% of its artists also exist on the Last.fm web site. This is a good example of very long tail content, for which traditional usage-based recommenders typically do not work so well [2].

In order to address the previous problems and achieve better recommendations than stand-alone techniques, usage and content-based approaches have been combined in many different ways as hybrid recommenders [1]. In this paper we describe the hybrid recommender system implemented on the Palco Principal and evaluate it. The hybrid recommender is compared against a usage- and a content-based recommender. We also propose performance measures to determine the impact of the recommenders in terms of user activity and loyalty.

2. DIFFERENT MODALITIES FOR ITEM SIMILARITIES

Item-based recommender systems exploit similarity among items [7]. The system looks into the set of items and computes the similarity between pairs of items, generating a matrix representing the similarities between all the pairs of items, according to a similarity measure. A representation of an item-item similarity matrix is shown below (each item i can be, for example, a music track).

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

[1] http://www.last.fm
[2] http://www.amazon.com
[3] http://www.pandora.com
[4] http://www.palcoprincipal.com

	i_1	i_2	$\cdots$	i_q
i_1	1	$sim(i_1, i_2)$	$\cdots$	$sim(i_1, i_q)$
i_2	$sim(i_2, i_1)$	1	$\cdots$	$sim(i_2, i_q)$
$\cdots$	$\cdots$	$\cdots$	1	$\cdots$
i_q	$sim(i_q, i_1)$	$sim(i_q, i_2)$	$\cdots$	1

The effectiveness of an item-based recommender system depends on the method used to calculate the similarity among the items in the matrix. Thus, in the next sections we present three different methods to calculate the similarity among music tracks. These methods tap into two different types of data: usage-based data on the one hand, and content-based data on the other hand.

2.1 Usage-based Similarity

The simplest form of usage data is a pair $< user, item >$ meaning that *user* had a positive interaction with *item* (e.g., user listened to a track). The positive nature of the interaction is often inferred from behavior. In the case of this work, we have access to playlists, which are collections of music tracks created and organized by individual users. The fact that a user adds a track to a playlist is regarded as a preference. Therefore a $< user, item >$ pair means, in our case, that the user added track (item) to his playlist and, ergo, likes this music. Usage data such as this is a particular case of preference data where each user rates some items on a given scale (e.g., 1 to 5). In this case, we have a binary scale (i.e., likes / does not like).

To compute the similarity between pairs of music tracks from usage data, for example, u_1 and u_2, we first isolate the users who have included the tracks in their playlists. Then, we compute the similarity $sim(u_1, u_2)$ between u_1 and u_2. In [7] the authors present three methods to measure similarity between pairs of items: cosine angle, Pearson's correlation and adjusted cosine angle. In this paper, we use the cosine angle, defined as

$$sim(u_1, u_2) = cos(\overrightarrow{u_1}, \overrightarrow{u_2}) = \frac{\overrightarrow{u_1}.\overrightarrow{u_2}}{||\overrightarrow{u_1}|| * ||\overrightarrow{u_2}||}, \qquad (1)$$

where $\overrightarrow{u_1}$ and $\overrightarrow{u_2}$ are binary vectors with as many positions as existing users. The value 1 means that the users included the track in their playlists. The value 0 is the opposite. The operator "." denotes the dot-product of the two vectors.

2.2 Tag-based Similarity

Social music tags are free text labels that are assigned to items such as artists, playlists and music tracks [5]. In our particular case, tags describe music tracks, and are typically words or short phrases related to genre, intrument and influence. For example, music tracks in our data are typically tagged with tags like *rock*, *guitar* or *Daft Punk*.

In order to capture the tag correlation, an $M \times N$ matrix of tracks and tags is built, where M is the number of tracks and N the number of tags. Matrix elements with values different than 0 mean that a given tag n has been used to annotate a given music track m. The rationale is that music tracks with similar tags would be similar. However, the dimensions M and N can be extremely large and the matrix very sparse, thus making the problem computationally expensive.

To overcome this problem, an information retrieval technique called Latent Semantic Analysis (LSA) [3] is used to analyze the inherent structure of the matrix. Basically, LSA makes use of Singular Value Decomposition (SVD) to create a space T of tag concepts by combining the N tags. This will reduce the dimensionality of the matrix by replacing the tags by a small number of tag concepts, while still preserving the similarity structure among rows or columns.

Then, we use the reduced matrix, $M \times T$, to calculate the similarity between pairs of music tracks. In our case, we use the cosine similarity distance, defined as

$$sim(t_1, t_2) = cos(\vec{t_1}, \vec{t_2}) = \frac{\vec{t_1} \cdot \vec{t_2}}{\left\|\vec{t_1}\right\| * \left\|\vec{t_2}\right\|}, \qquad (2)$$

where $\vec{t_1}$ and $\vec{t_2}$ are binary vectors with all the tag concepts. A value of 1 or 0 represents the presence or absence, respectively, of the tag concept for the given music track.

2.3 Audio-based Similarity

For this approach, we have used the free MARSYAS framework[5] to extract 16 audio features from 46ms frames of the audio signals with no overlap. The features are: the spectral centroid, rolloff frequency, spectral flux, and 13 MFCCs, including MFCC0 [8]. Features are aggregated in 1s texture windows, and then averaged over the whole file. Final features are average and standard deviation. We chose these features because of availability of code.

After extracting the audio features for each track, we calculate the similarity among the tracks. The similarity is calculated by the Euclidian distance through the 16 audio features. Here, we define the Euclidian distance between 2 tracks, a_1 and a_2, as follows

$$sim(a_1, a_2) = euclidian(\overrightarrow{a_1}, \overrightarrow{a_2}) = \sqrt{\sum_{f=1}^{16} (\overrightarrow{a_{1_f}} - \overrightarrow{a_{2_f}})^2}, \qquad (3)$$

where $\overrightarrow{a_1}$ and $\overrightarrow{a_2}$ are vectors with the 16 audio features.

Note that contrarily to the cosine, where the similarity is directly related to the measure, with the Euclidian distance the similarity is inversely related to the measure, i.e., the lower the measure the higher the similarity.

3. MUSIC RECOMMENDATION BASED ON DIVERSE MODALITIES

In this section, we show how the similarity methods presented in Section 2 can be used to recommend music tracks. We start by describing a usage- and a content-based recommender system, which are used as benchmark systems in this paper. Then, we propose a hybrid recommender system that combines both usage and content.

3.1 Usage-based Recommendation

Usage-based recommendation is made on the basis of the similarity matrix between tracks described in Section 2.1. Given a user, his playlists are merged and the music tracks in it are used as seeds (S) for the recommendations. The general procedure follows the Item-based Collaborative Filtering algorithm [7]. For each recommendable music track m we fetch its k closest neighbors $N(m)$. From the seeds $s \in S$, the neighbors and their similarities we calculate the activation weight *ActWeight* of each track m which is not already in the playlists of the user.

[5] `http://marsyas.info`

$$ActWeight(m) = \frac{\sum\limits_{s \in N(m) \cap S} sim(m,s)}{\sum\limits_{n \in N(m)} sim(m,n)}. \qquad (4)$$

Note that we exclude for recommendation tracks that are already in the playlist.

In our music recommendation application, we also have a source of negative information, called blacklist (B). When recommendations are shown to the user, he has the option of blacklisting a particular recommendation. This way, the blacklisted track is not shown again. Here, we exclude from the similarity matrix the tracks in the blacklist B of the seed user. Moreover, the blacklist information is used to calculate a global acceptance index $AccI$ of each track. This index captures the tendency of a track for being blacklisted and is calculated from the number of times a track is blacklisted $B(m)$ and the number of times it is included in a playlist $P(m)$. The value 1 means that the track is not included in any blacklist.

$$AccI(m) = 1 - \frac{B(m)}{B(m) + P(m) + 1}. \qquad (5)$$

After calculating $AccI(m)$ it is multiplied by $ActWeight(m)$ to obtain the final score of the track. Recommended tracks are then sorted by score from highest to lowest.

3.2 Content-based Recommendation

The content-based recommender system that we describe in this section combines tags and audio features to recommend music tracks. As proposed in [2], audio features should be good for low-level similarities (e.g., the main timbre of music tracks), while tags should be good supplements as they account for higher-level information that could not be reliably computed from audio (e.g., *female voice*).

The system starts by computing two item-item similarity matrices (Section 2). One matrix is computed using tags (Section 2.2) and the other one using audio features (Section 2.3). Once we have the two matrices, we can generate the recommendations. Given a seed music track, $s \in S$, the system first fetches its k closest neighbors on each matrix, generating two lists of recommendable music tracks, i.e., one based on tags and the other based on audio features. Then, the system ranks each list separately, taking into account the similarities, and computes a final ranking where the position is the sum of the two ranks in every independent ranking. Finally, the k best ranked music tracks, according to the final ranking, are recommended.

3.3 Recommendation Combining Usage and Content

The recommendation strategy that combines Usage and Content data, referred to as **Mix**, is described in this section. Given a user playlist, we produce three lists of k recommendations. One obtained from usage data (R_u), one from tags (R_t) and the third from audio data (R_a). These three lists are sorted by inverse order of relevance of the recommendations. For each list, the recommended tracks are assigned ranks from k (top recommendation) to 1. The combined rank for each track is the average of the three ranks. For example, if a track m is the first recommendation in R_u, second in R_t and does not occur in R_a, and assuming $k = 100$, the combined rank is $(100 + 99 + 0)/3 = 66.33$.

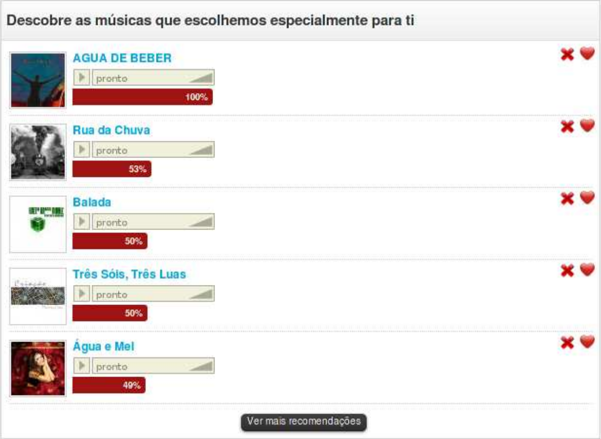
The blacklist information is also used by multiplying the combined rank by the $AccI$ (equation 5) of the track to be recommended. This will penalize tracks that are blacklisted by a large number of users.

4. CASE STUDY

The recommendation strategies described in the previous section have been deployed on Palco Principal, a start-up company that holds a web site of Portuguese music since 2007. Besides music recommendations, the site also provides services like news, advertisements, social networking and an application for users to access the services of the site through their mobile phone.

During the period of our study, the site had about 76000 users (61223 listeners and 14777 artists who uploaded music) and 61000 music tracks. From the tags available in the site, we used 373 tags which can be categorized into three classes: genre (e.g., hip hop), intrument (e.g., clarinet) and influence (e.g., Daft Punk). There is a minimum of 1, a mean of 3.52 and a maximum of 36 tags per track. As already stated, this content corresponds to the very end of the long tail of music [2].

In the site, each of the recommenders are used separately. When a user opens the page for managing playlists, the recommender is invoked in real time and the results are shown to the user (Figure 1). The user can then listen to recommended tracks, add tracks to his playlist (heart) or add tracks to his blacklist (cross).



Descobre as músicas que escolhemos especialmente para ti

AGUA DE BEBER

Rua da Chuva

Balada

Três Sóis, Três Luas

Água e Mel

Figure 1: Recommendations as shown to the user.

4.1 Evaluation Methodology

To compare the merits of the three recommenders (**Usage**, **Content** and **Mix**) we have performed an online evaluation [4] and followed the reactions of users during 22 weeks, between 10/20/2010 and 03/22/2011. These were real users with no knowledge of the evaluation in course. Each new user was assigned one of the three recommenders. The assignment was decided by the remainder of the division of the user ID by 3. This way, we had a random assignment of users to each of the recommenders, and the same user would always get recommendations from the same source.

User activity has been recorded in two different ways. One was Google Analytics (GA) and the other was the site's internal data base (DB). In the case of GA, we have associated events to user actions of adding to playlist and adding to blacklist. In the case of DB, we have the playlist and blacklist tables in the data base. To be able to identify whether each track added to the playlists had been recommended, we added a source field indicating which recommender had done the job. In the end, we have observed some non significant differences in the values obtained from GA and DB, which comforted us in the quality of the data to be analyzed.

To measure the variation of the recommenders effects, we have divided the 22 weeks into 11 periods of 2 weeks. For each period we have measured the number of sessions (S), the number of additions to playlists (P) and the number of additions to blacklists (B) for each recommender.

From these three basic measures we have defined the following derived measures:

$$\begin{aligned} \textit{Activity rate} &= (P+B)/S, & (6)\\ \textit{Absolute acceptance rate} &= P/S, & (7)\\ \textit{Relative acceptance rate} &= P/(P+B). & (8) \end{aligned}$$

Google Analytics also provides information about the number and frequency of users who return to the site. For a given period, $L(x)$ is the number of users who return x times to the site. Loyalty can then be measured in many different ways. We have tried to capture loyalty by counting users who return three or more than three times and using as reference the number of users who return less than three times. We call this measure Loyalty3 rate.

$$\textit{Loyalty3 rate} = \frac{\sum_{x \geq 3} L(x)}{L(1) + L(2)}. \qquad (9)$$

For each measure, and each recommender, we have collected samples with values from the 11 periods. We then compare averages and standard deviations of the measures and perform two-tailed t-tests ($\alpha = 0.05$) to determine the significance of the differences.

4.2 Results

In this section we discuss the results obtained with our case study. During the evaluation period there were about 57000 sessions involving recommendations, where 1327 users made 3267 additions to playlists and 3123 additions to blacklists.

As reported in Table 1, overall, **Mix** shows a slightly lower relative acceptance rate (RAR) than **Content** and **Usage**. However, the differences are not significant (this is due to the high variability of all three recommenders with respect to the 11 periods of 2 weeks, as shown in the relatively high standard deviations), and all three recommenders have a relative acceptance around 0.5. This can be understood as follows: in response to a given recommendation, the user is as likely to react with an addition to playlist (i.e., a positive reaction) than an addition to blacklist (i.e., a negative reaction). This appears to be true for all three recommenders.

This does not however mean that the three recommenders have a similar performance. Indeed, given a recommendation, a user can not only react by an addition to playlist or to blacklist, but also not react at all –which in our opinion is another negative reaction. As can be seen in Table 1, activity rate (AR) measure, our data shows that for the same number of recommendations, the **Mix** recommender results in more user activity than the other two. In other words, it appears that users are more likely to react to recommendations when confronted with recommendations of **Mix** than those of the other two. This means that users will generate more additions to playlist, and more additions to blacklist, with **Mix** than with **Content** and **Usage**. The former can be observed in Table 1, that show a significantly higher absolute acceptance rate (AAR) for **Mix**. When compared to **Content**, **Mix** presents a gain of 119%. With respect to the **Usage**, it shows a gain of 50%. This means that users getting the **Mix** suggestions had a significant tendency for reacting more positively to recommendations.

Finally, regarding the loyalty3 rate (L3R), we see in Table 1 that the **Mix** recommender is similar to **Content** but significantly better than **Usage**. There, the system **Mix** presents a gain of 16% when compared to **Usage**.

Table 1: Results. Values with (*) represent recommendation methods whose differences with Mix are statistically significant (p-value $<$ 0.05).

Measure	System	Mean	Std. Dev.	p-value
RAR	Mix	0.499	0.157	-
	Content	0.512	0.164	0.848
	Usage	0.600	0.125	0.162
AR	Mix	**0.165**	0.061	-
	Content	0.074 (*)	0.025	0.001
	Usage	0.088 (*)	0.021	0.002
AAR	Mix	**0.081**	0.038	-
	Content	0.037 (*)	0.018	0.013
	Usage	0.054 (*)	0.023	0.049
L3R	Mix	**1.880**	0.376	-
	Content	1.870	0.171	0.867
	Usage	1.620 (*)	0.196	0.044

5. CONCLUSIONS

In this paper we proposed a music recommender system that combines usage and content data. Our proposal has been evaluated online, with real users, on a commercial web site of music from the very long tail. Our results show that **Mix** generates more activity and at least the same amount of positive responses of **Content** and **Usage** (or more, depending on the evaluation measure). Finally, it has good results in terms of promoting user loyalty. Hence our conclusion that **Mix** tends to be a better option to music recommendation than the other two. **Mix** is currently the core recommendation engine on `http://www.palcoprincipal.com`.

We are currently developing a monitoring tool for continuously collecting and analyzing the activity of the recommenders of the site. This will allow the owners of the site to keep an eye on the impact of the recommenders. On the other hand, it will give us more reliable data and will enable us to look into other facets of the recommendations, such as variety and sensitivity to the order. With that information we will be able to better understand what makes users more active, as well as to design recommenders that may have different mixes, depending on the profile of the user.

6. ACKNOWLEDGMENTS

This work is supported by the QREN-AdI Palco3.0/3121 PONORTE; the European Commission, FP7 (Seventh Framework Programme), ICT-2011.1.5 Networked Media and Search Systems, grant agreement No 287711; and the ERDF through the Programme COMPETE and by the Portuguese Government through FCT - Foundation for Science and Technology, project ref. PTDC/EAT-MMU/112255/2009.

7. ADDITIONAL AUTHORS

Additional authors: Mohamed Sordo (Universitat Pompeu Fabra, Spain, email: `mohamed.sordo@upf.edu`).

8. REFERENCES

[1] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12:331–370, 2002.

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

[3] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41:391–407, 1990.

[4] R. Kohavi, R. Longbotham, D. Sommerfield, and R. M. Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, 2009.

[5] P. Lamere. Social tagging and music information retrieval. *Journal of New Music Research*, 37(2):101 – 114, 2008.

[6] P. Ruppert, R. Hart, and S. Evans. The 2007 digital music survey, Entertainment Media Research, 2007. `http://www.slideshare.net/patsch/emr-digital-music-survey-2007`.

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Tenth International Conference on World Wide Web*, pages 285–295, Hong Kong, 2001.

[8] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293 – 302, 2002.