Online Recommender System for Radio Station Hosting

Dmitry I. Ignatov[1], Andrey Konstantinov[1], Sergey Nikolenko[2,3], Jonas Poelmans[1], and Vasily Zaharchuk[1]

[1] National Research University Higher School of Economics
dignatov@hse.ru
http://www.hse.ru
[2] Steklov Mathematical Institute, St. Petersburg, Russia
[3] St. Petersburg Academic University, St. Petersburg, Russia

Abstract. We describe a new recommender system for the Russian interactive radio network FMhost. The underlying model combines collaborative and user-based approaches. The system extracts information from tags of listened tracks for matching user and radio station profiles and follows an adaptive online learning strategy based on user history. We also provide some basic examples and describe the quality of service evaluation methodology.

Keywords: music recommender systems, interactive radio network, e-commerce, quality of service

1 Introduction and related work

Music recommendation is an important topic in the field of recommender systems. Recent works in this area can be found in the proceedings of the International Society for Music Information Retrieval Conference (ISMIR) [1], the Workshop on Music Recommendation and Discovery (WOMRAD) [2,3], and the Recommender Systems conference (RecSys) [4]. Several broadcasting services including LastFm, Yahoo!LaunchCast and Pandora are well known and work on a commercial basis. The latter two of them do not broadcast for Russia. Despite the many high-quality papers on different aspects of music recommendation, there are only few studies devoted to online radio station recommender systems.

This work is devoted to the Russian online radio hosting service FMhost and, in particular, its new hybrid recommender subsystem. Recently, the focus of computer science research for the music industry has shifted from music information retrieval and exploration [5,6,7] to music recommender services [8,9]. The topic is not new (see, e.g., [10]); however, it is now inspired by new capabilities of large online services to provide not only millions of tracks for listening to, but even radio station hosting. Social tagging is also one of the important factors which allows to apply new tag-similarity based recommender algorithms to the domain [11,12].

Recently, a widely acclaimed public contest on music recommender algorithms, KDD Cup, was held by Yahoo! (`http://kddcup.yahoo.com/`). In KDD Cup, track 1 was devoted to learning to predict users' ratings of musical items (tracks, albums, artists and genres) in which items formed a taxonomy. Each track belonged to an album, albums belonged to artists, and together they were tagged by genres. Track 2 aimed at developing learning algorithms for separating music tracks scored highly by specific users from tracks not scored by them. It attracted a lot attention from the community to problems which are both typical for recommender systems and specific for music recommendation: scalability issues, capturing the dynamics and taxonomical properties of items [13]. The current trends of music recommender systems reflect advantages of hybrid approaches and show the need for user-centric quality measures [14]. For instance, in [15] an interesting approach based on a "forgetting curve" to evaluate "freshness" of predictions was proposed. In [16], the authors posed an important question, namely how much metadata do we need in music recommendation, and after a subjective evaluation of 19 users the authors concluded that pure content-based methods can be drastically improved by using genres.

In [17], the authors proposed the music recommender system Starnet for social networking. It generates recommendations based either on positive ratings of friends (social recommendations), positive ratings of others in the network (non-social recommendations), and it also makes random recommendations. Another interesting online music recommendation system we can mention is Hotttabs [18], dedicated to guitar learning. Some authors aim at improving music recommender systems by using semantic extraction techniques [19,20]. In [21] the author describes a system of genre recommendation for music and TV programs, which can be considered as an alternative channel selector. The authors of [22] proposed a recommender system GroupFan which is able to aggregate preferences of group users to their mutual satisfaction.

Many online services (e.g., Last.fm or LaunchCast) call their audio streams "radio stations", but in reality they produce a playlist from a database of tracks based on a recommender system rather than actually recommend a radio channel. FMhost, on the other hand, provides users with online radio stations in the classical meaning of this term: there are human DJs who perform live, a radio station actually represents a strategy or mood of a certain person (DJ), they play their own tracks, perform contests etc. Thus, the problem we are solving differs from most of the work done in music recommendation, and some of the challenges are unique.

The paper is organized as follows. In Section 2, we describe our online radio service FMhost. In section 3, we propose our new recommender model, two basic recommender algorithms, and describe the recommender system architecture. Quality of Service (QoS) measurement for the system and some insights on FMhost user behaviour are discussed in Section 4. Section 5 concludes the paper.

2 Online service FMhost.me

2.1 A concise online broadcasting dictionary

Before we proceed, we need to shortly explain some basic domain terminology.

A *chart* is a radio station track rating; for example, the rock chart shows a certain number (say, 10) of most popular rock tracks, ranked from the most popular (rank 1) to the least popular (rank 10) according to the survey. A *live performance* (or just *live* for short) is a performance to which one or several *DJs* (*disk jockeys*) are assigned. They do it from their own PCs, and the audio stream is being redirected from them to the Icecast server and then everywhere. Also they may have their own blog for each live, where people may interact with DJs who perform live. *LiquidSoap* is a sound generator that broadcasts audio files (*.mp3, *.aac etc.) into an audio stream. *Icecast* is a retranslation server that redirects an audio stream from one source, for example LiquidSoap, to many receivers.

2.2 The FMhost project

FMhost is an interactive radio network. This portal allows users to listen and broadcast their own radio stations. There are four user categories in the portal: (1) unauthorized user; (2) listener; (3) Disk Jockey (DJ); (4) radio station owner.

User capabilities vary upon their status. Unauthorized listeners can listen to any station, but they cannot vote or become DJs. They also cannot use the recommender system and the rating system.

Listeners, unlike unauthorized users, can vote for tracks, lives, and radio stations. They can use a recommender system or rating system. They can subscribe to lives, radio stations, or DJs. They also can be appointed to a live and become a DJ.

There are three types of broadcasting: (1) stream redirection from another server; (2) AutoDJ translation; (3) live performance.

Stream redirection applies when a radio station owner has its own server and wants to use FMhost as a broadcasting platform, but also wants to broadcast using his own sound generator, e.g., SamBroadcaster (`http://spacial.com/sam-broadcaster`), LiquidSoap (`http://savonet.sourceforge.net/`) etc. AutoDj is a special option that allows the users to play music directly from the FMhost server. Every radio owner gets some space where he can download as much tracks as he can, and then LiquidSoap will generate the audio stream and the Icecast (`http://www.icecast.org/`) server will redirect it to the listeners. Usually the owner sets a radio schedule which is being played.

Live performances are done by DJs. Everyone who has performed live at least once can be called a DJ. He can also be added to a radio station crew. Moreover, a DJ can perform lives at any station, not only on his own station where he is in a crew.

FMhost was the first project of its kind in Russia, starting in 2009. Nowadays, following FMhost's success, there exist several radio broadcasting portals, such as http://frodio.com/, http://myradio24.com/, http://myradio24.com/, http://www.radio-hoster.ru/, http://www.taghosting.ru/, http://www.economhost.com/, and even http://fmhosting.ru/. In late 2011, FMhost was taken down for a serious rewrite of the codebase and rethinking of the recommender system's architecture. In this paper, we describe the results of this upgrade.

The previous version of the recommender system experienced several problems, such as tag discrepancy or personal tracks without tags at all. A survey by FMhost with about a hundred respondents showed that more than half of them appreciated the previous version of our recommender system and more than 80% of the answers were positive or neutral (see Table 1); nevertheless, we hope that the new recommender model and algorithms provide even more accurate recommendations and make even less prediction mistakes.

Table 1. FMhost's recommender system satisfaction survey.

User opinion	Number of respondents (%)
I like it very much, all recommendations were relevant	54 (49%)
Good, I like most of the radio stations	22 (20%)
Sometimes there are interesting stations	16 (14%)
I like only few recommended radio stations	9(8%)
None of the recommended stations was satisfactory	10 (9%)

2.3 FMhost conceptual improvements

The new version features a more complex system of user interaction. Every radio station has an owner who is not just a name but also has the ability to assign DJs for lives, prepare radio schedule, and assign lives and programs. There will be a new broadcasting panel for DJs that will allow them to play tracks with additional features that were not available before, such as an equalizer or fading between tracks. A new algorithm for the recommender system, a new rating system, and a new chart system will be launched.

The rating system has been developed to rank radio stations and DJs according to their popularity and quality of work. A new core is being implemented and a new concept of LiquidSoap and Icecast is being designed. The system is designed such that all problems that have surfaced in the previous version were eliminated.

3 Models, algorithms and recommender architecture

3.1 Input data and general structure

Our model is based on three data matrices. The first matrix $A = (a_{ut})$ tracks the number of times user u visits radio stations with a certain tag t. Each radio station r broadcasts audio tracks with a certain set of tags T_r. The sets of all users, radio stations, and tags are denoted by U, R, and T respectively. The second matrix $B = (b_{rt})$ contains how many tracks with a tag t a radio station r has played. Finally, the third matrix $C = (c_{ur})$ contains the number of times a user u visits a radio station r. For each of these three matrices, we denote by v^A, v^B, and v^C the respective vectors containing sums of elements: $v^A = \sum\limits_{t \in T} a_{ut}$, $v^B = \sum\limits_{t \in T} b_{rt}$, and $v^C = \sum\limits_{r \in R} a_{ur}$. We also denote for each matrix A, B, C the corresponding frequency of visits matrix by A_f, B_f, and C_f; the frequency matrix is obtained by normalizing the matrix with the respective visits vector, e.g., $A_f = (a_{ut} \cdot (v^A_u)^{-1})$. Our model is not purely static; the matrices A, B, and C change after a user u visits a radio station r with a tag t, i.e., each value a_{ut}, b_{rt}, and c_{ur} is incremented by 1 after this visit.

The model consists of three main blocks: the Individual-Based Recommender System (IBRS) model, the Collaborative-Based Recommender System (CBRS) model, and the End Recommender Systems (ERS) that aggregates the results of the former two.

Each model has its own algorithmic implementation. Since both our previous works [23,24] and this work implicitly use biclustering ideas, we continue to name our general algorithms with the RecBi acronym; this time it is the RecBi3 family. We call the resulting algorithms for the three proposed models RecBi3.1, RecBi3.2, and RecBi3.3, respectively. Here we do not use the notation from formal concept analysis, but refer to [25] for the basic notation used in our previous algorithms RecBi2.1 and RecBi2.2.

3.2 IBRS

The **IBRS** model uses matrices A_f and B_f and aims to provide a particular user $u_0 \in U$ with top N recommendations represented mathematically by a special structure $Top_N(u)$. Formally, $Top_N(u_0)$ is a triple $(R_{u_0}, \preceq_{u_0}, \mathrm{rank})$, where R_{u_0} is the set of at most N radio stations recommended to a particular user u_0, $\preceq_{u_0}$ is a well-defined quasiordering (reflexive, transitive, and complete) on the set R_{u_0}, and rank is a function which maps each radio station r from R_{u_o} to $[0, 1]$.

The **RecBi3.1** algorithm computes the 1-norm distance between a user u_0 and a radio station r, i.e.m $d(u_0, r) = \sum t \in T|a_{u_0t} - b_{rt}|$. Then all distances between the user u_0 and the radio stations $r \in R$ are calculated. Further the algorithm constructs the relation $\prec_{u_0}$ according to the following rule: $r_i \preceq r_j$ iff $d(u_0, r_i) \leq d(u_0, r_i)$. The function rank operates on R_{u_0} according to the following rule:

$$\mathrm{rank}(r_i) = 1 - d(u_0, r_i)/\max_{r_j \in R} d(u_0, r_j).$$

Finally, after selecting N radio stations for N greatest rank values in the set R_{u_0}, we have the structure $Top_N(u_0)$ which represents a ranked list of radio stations recommended to the user u_0.

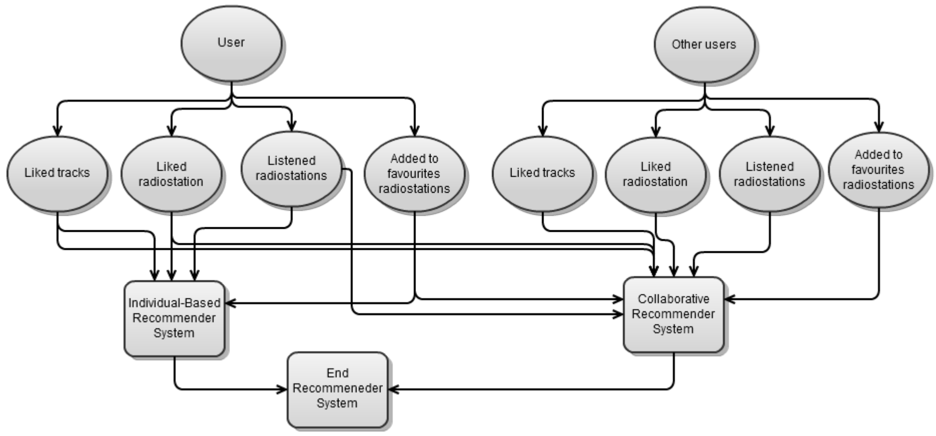

Fig. 1. The recommender system architecture

As shown in Fig. 1, our model takes into account not only "listened tracks" but also "liked tracks", "liked radio stations", and "favorite radio stations". To refine the IBRS submodel we tune it with the SMARTS algorithm known from decision making theory [26]. According to the method and expert decisions, we should count each track tag of a "listened radio station", "liked radio station", "liked track", and "favorite radio station" with a different weight. The SMARTS procedure provided us with the four weights for "listened radio station", "liked radio station", "favorite radio station", and "liked track" according to our experts' assessment of mutual criterion importance, namely 0.07, 0.16, 0.3, and 0.47. In the SMARTS method, we consider each tag type as a criterion with two terminal values 0 and 100% on a real number scale. Some tag t may have some or even four of these types simultaneously; in this case, the algorithm adds to a_{ut} the total weight of the tag (i.e., the sum of weights) after a user u visits some radio station with this tag. In case there are several elements with the same rank so that $Top_N(u)$ is not uniquely defined, we simply choose the first elements according to some arbitrary ordering (e.g., the lexicographic ordering of station names).

3.3 CBRS

The **CBRS** model is based on the C_f matrix. The matrix also yields a vector n^C which stores the total number of listened stations for each user $u \in U$. This vector also changes over time, and this value is used as a threshold to transform matrix C_f to distance matrix D as follows:

$$d_{ij} = \begin{cases} |c_{fir} - c_{fjr}|, & if\ c_{fir} \geq n_i^{-1} \text{and } c_{fjr} \geq n_j^{-1} \\ |c_{fir} + c_{fjr}|, & if\ c_{fir} > n_i^{-1} \text{and } c_{fjr} < n_j^{-1} \text{or vice versa} \end{cases} \qquad (1)$$

This distance takes into account the frequency n_u^C of all radio station visits for user u and considers its inverse value as a threshold to decide whether a particular station r should be considered as popular for this user. Thus, users with different signs of $c_{fir} - n_i^{-1}$ and $c_{fjr} - n_j^{-1}$ become more distant than for the conventional absolute distance. This distance d_{ij} actually serves as a sort of polarizing filter, and in Section 4 we compare it with common approaches.

After computing D, the algorithm **RecBi3.2** constructs the list $Top_k(u_0) = (U_{u_0}, \preceq_{u_0}, \text{sim})$ of k users similar to our target user u_0 who awaits recommendations, where $\text{sim}(u) = 1 - d_{uu_0} / \max\limits_{u' \in U} d_{u'u_0}$. We define the set of all radio stations user u_0 listened to as $L(u_0) = \{r | c_{fur} = 0\}$. In a similar way, we define

$$\begin{aligned} Top_N(u_0) &= (R_{u_0}, \preceq_{u_0}, \text{rank}), \text{ where} \\ \text{rank}(r) &= \text{sim}(u^*) \cdot c_{fu^*r} \text{ and} \\ u^* &= \arg \max_{u \in U_{u_0}, r \in U/L(u_0)} \text{sim}(u) \cdot c_{fur}. \end{aligned}$$

It is worth mentioning that $\text{rank} : r \mapsto [0,1]$. The problem of choosing exactly N topmost stations is solved in the same way as in the IBRS submodel.

3.4 ERS

After IBRS and CBRS have finished, we are left with two ranked lists of recommended stations $Top_N^I(u_0)$ and $Top_N^C(u_0)$ for our target user u_0 from IBRS and CBRS respectively. The **ERS** submodel proposes a simple solution for aggregating these lists into the final recommendation structure $Top_N^E(u_0) = (R_{u_0}^E, \preceq_{u_0}^E, \text{rank}^E)$. For every $r \in R_{u_0}^C \cup R_{u_0}^I$, the function $\text{rank}^E(r)$ maps r to the weighted sum

$$\beta \cdot \text{rank}^C(r) + (1 - \beta) \cdot \text{rank}^I(r),$$

where we let $\beta \in$ [0,1], $\text{rank}^C(r) = 0$ for all $r \notin R^C$ and $\text{rank}^I(r) = 0$ for all $r \notin R^I$. The algorithm **RecBi3.3** adds the best N radio stations according to this criterion to the set $R_{u_0}^C$.

4 Quality of service assessment

To evaluate the quality of the developed system, we propose a variant of the cross-validation technique [27]. Before we proceed to the detailed description of the procedure, we discuss some important analyses that we conducted on the FMhost data for the period from 2009 till 2011.

4.1 Basic statistics

It is a well-known fact that social networking data often follows the so called power law distribution [28]. To decide which amount of active users or radio stations we have to take into account for making recommendations, we performed a simple statistical analysis of user and radio station activity. Around 20% of the users (only registered ones) were analysed.

Table 2. Basic parameters of the user and radio visits datasets, along with their power-law fits and the corresponding *p-value* .

Dataset	n	$\langle x \rangle$	σ	x_{max}	$\hat{x}_{min}$	$\hat{\alpha}$	n_{tail}	*p-value*
User dataset	4187	5.86	12.9	191	12 ± 2	2.46(0.096)	117	**0.099**
Radio dataset	2209	11.22	60.05	1817	46 ± 11	2.37(0.22)	849	**0.629**

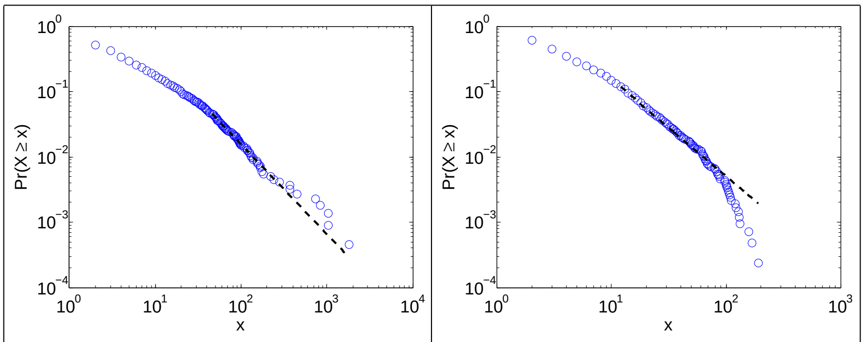

Fig. 2. Cumulative distribution functions $P(x)$ and their maximum likelihood power-law fits for the FMhost two empirical data sets. (a) The frequency distribution of radio station visits. (b) The frequency of visits of unique users.

Table 2 shows p-values of statistical tests, which were performed by means of Matlab tools from [28], show that the power law does fit the radio station dataset, and the probability to make an error by ruling out the null hypothesis (no power law) is about 0.1 for the user dataset. Thus, the radio station visits dataset is more likely to follow the power law than the user visits dataset, but we should take it into account for both datasets; Fig. 2 shows how the power law actually fits our data.

This analysis implies useful consequences according to the well-known “80:20” rule:

$$W = P^{(\alpha-2)/(\alpha-1)},$$

which means that the fraction W of the wealth is in the hands of the richest P of the population. In our case, 50% of users make 80% of all radio station visits, and 50% of radio stations have 83% of all visits. Thus, if the service tends to take into account only active stations and users, it can cover 80% of all visits by considering only 50% of their active audience. However, new radio stations still deserve to be recommended, so this rule can only be applied to the user database.

4.2 Quality assessment

To evaluate QoS for the IBRS subsystem (RecBi3.1 algorithm), we count average precision and recall on the set $R_N \subset R$, where N is a number of randomly "hidden" radiostations. We suppose that for all r in R_N and every user $u \in U$ the algorithm does not know whether the radio stations were liked, added to favorites, or even visited, and we change A_f and R accordingly. Then RecBi3.1 attempts to recommend Top-N radio stations for this modified matrix A_f.

Top-N average precision and recall are computed as follows:

$$\text{Precision} = \frac{\sum\limits_{u \in U} \frac{|R_u^I \cap L_u \cap R_N|}{|L_u \cap R_u^I|}}{|U|},$$

$$\text{Recall} = \frac{\sum\limits_{u \in U} \frac{|R_u^I \cap L_u \cap R_N|}{|L_u \cap R_N|}}{|U|}.$$

To deal with CBRS, we use a modification of the leave-one-out technique. At each step of the procedure for a particular user u, we "hide" all radio stations $r \in R_N$ by setting $c_{fur} = 0$. Then we perform RecBi3.2 assuming that $c_{fu'r}$ is unchanged for $u' \in U/u$. After that we compute

$$\text{Precision} = \frac{\sum\limits_{u \in U} \frac{|R_u^C \cap L_u \cap R_N|}{|L_u \cap R_u^C|}}{|U|},$$

$$\text{Recall} = \frac{\sum\limits_{u \in U} \frac{|R_u^C \cap L_u \cap R_N|}{|L_u \cap R_N|}}{|U|}.$$

To tune the ERS system, we can use a combination of these two procedures trying to find the optimal β as

$$\beta^* = \arg\max_{\beta} \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{(\text{Precision} + \text{Recall})}$$

.

We suppose that in one month of active operation we will have enough statistics to tune β and choose appropriate similarity and distance measures as well as thresholds. We suppose that the resulting system will provide reasonably accurate recommendations using only a single (last) month of user history and only 50% of the most active users. For quality assessment during the actual operation, we will compute Top-3, Top-5, and Top-10 Precision and Recall measures as well as whether the system provides a user only with Top-10 items with a highest rank. In addition, online surveys can be launched to assess user satisfaction with the new RS system.

5 Conclusion and further work

In this work, we have described the underlying models, algorithms, and the system architecture of the new improved FMhost service. We hope that the developed algorithms will help a user to find relevant radio stations to listen to. In future optimization and tuning, special attention should be paid to scalability issues and user-centric quality assessment. We consider matrix factorization techniques as a reasonable tool to increase scalability, but it has to be carefully adapted and assessed taking into account the folksonomic nature of tracks tags. Another attractive feature of the developed system is that it can serve as a kind of World of Music map built on track-to-track similarity matrices with tags [7]. Another important issue is dealing with the triadic relational nature of data (users, radio stations (tracks), and tags), which constitutes the so called *folksonomy* [29], a primary data structure in tagging resource-sharing systems. As shown in [30], this data can be successfully mined by means of triclustering, so we also plan to build a tag-based recommender system by means of triclustering.

Acknowledgments. We would like to thank Rustam Tagiew and Mykola Pechenizkiy for their comments, remarks and explicit and implicit help during paper preparations. The work of Sergey Nikolenko has been supported by the Russian Foundation for Basic Research grant 12-01-00450-a, the Russian Presidential Grant Programme for Young Ph.D.'s, grant no. MK-6628.2012.1, for Leading Scientific Schools, grant no. NSh-3229.2012.1, and RFBR grants 11-01-12135-ofi-m-2011 and 11-01-00760-a. The study was implemented in the framework of the Basic Research Program at the National Research University Higher School of Economics in 2012 and in the Laboratory of Intelligent Systems and Structural Analysis.

References

1. Klapuri, A., Leider, C., eds.: Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011. In Klapuri, A., Leider, C., eds.: ISMIR, University of Miami (2011)
2. Anglade, A., Baccigalupo, C., Casagrande, N., Celma, Ò., Lamere, P.: Workshop report: Womrad 2010. In Amatriain, X., Torrens, M., Resnick, P., Zanker, M., eds.: RecSys, ACM (2010) 381–382

3. Anglade, A., Celma, O., Fields, B., Lamere, P., McFee, B.: Womrad: 2nd workshop on music recommendation and discovery. In: Proceedings of the fifth ACM conference on Recommender systems. RecSys '11, New York, NY, USA, ACM (2011) 381–382
4. RecSys '11: Proceedings of the fifth ACM conference on Recommender systems, New York, NY, USA, ACM (2011) 609116.
5. Hilliges, O., Holzer, P., Klüber, R., Butz, A.: Audioradar: A metaphorical visualization for the navigation of large music collections. In Butz, A., Fisher, B.D., Krüger, A., Olivier, P., eds.: Smart Graphics. Volume 4073 of Lecture Notes in Computer Science., Springer (2006) 82–92
6. Gleich, D.F., Rasmussen, M., Lang, K., Zhukov, L.: The world of music: User ratings; spectral and spherical embeddings; map projections. Online report (2006)
7. Gleich, D.F., Zhukov, L., Rasmussen, M., Lang, K.: The World of Music: SDP Embedding of High Dimensional data. In: Information Visualization 2005. (2005) Interactive Poster.
8. Brandenburg, K., Dittmar, C., Gruhne, M., Abeer, J., Lukashevich, H., Dunker, P., Grtner, D., Wolter, K., Grossmann, H.: Music search and recommendation. In Furht, B., ed.: Handbook of Multimedia for Digital Entertainment and Arts. Springer US (2009) 349–384
9. Celma, Ò.: Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space. Springer (2010)
10. Avesani, P., Massa, P., Nori, M., Susi, A.: Collaborative radio community. In: Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. AH '02, London, UK, UK, Springer-Verlag (2002) 462–465
11. Symeonidis, P., Ruxanda, M.M., Nanopoulos, A., Manolopoulos, Y.: Ternary semantic analysis of social tags for personalized music recommendation. In Bello, J.P., Chew, E., Turnbull, D., eds.: ISMIR. (2008) 219–224
12. Nanopoulos, A., Rafailidis, D., Symeonidis, P., Manolopoulos, Y.: Musicbox: Personalized music recommendation based on cubic analysis of social tags. IEEE Transactions on Audio, Speech & Language Processing **18**(2) (2010) 407–412
13. Koenigstein, N., Dror, G., Koren, Y.: Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In: Proceedings of the fifth ACM conference on Recommender systems. RecSys '11, New York, NY, USA, ACM (2011) 165–172
14. Celma, O., Lamere, P.: Music recommendation and discovery revisited. In: Proceedings of the fifth ACM conference on Recommender systems. RecSys '11, New York, NY, USA, ACM (2011) 7–8
15. Hu, Y., Ogihara, M.: Nextone player: A music recommendation system based on user behavior. [1] 103–108
16. Bogdanov, D., Herrera, P.: How much metadata do we need in music recommendation? a subjective evaluation using preference sets. [1] 97–102
17. Mesnage, C.S., Rafiq, A., Dixon, S., Brixtel, R.P.: Music discovery with social networks. In: Workshop on Music Recommendation and Discovery 2011, Workshop on Music Recommendation and Discovery (October 2011) 1–6
18. Barthet, M., Anglade, A., Fazekas, G., Kolozali, Sefki Macrae, R.: Music recommendation for music learning: Hotttabs, a multimedia guitar tutor. In: Workshop on Music Recommendation and Discovery 2011, Workshop on Music Recommendation and Discovery (October 2011) 7–13

19. Tatl, I., Birturk, A.: Using semantic relations in context-based music recommendations. In: Workshop on Music Recommendation and Discovery 2011, Workshop on Music Recommendation and Discovery (October 2011) 14–17
20. Knees, P., Schedl, M.: Towards semantic music information extraction from the web using rule patterns and supervised learning. In: Workshop on Music Recommendation and Discovery 2011, Workshop on Music Recommendation and Discovery (October 2011) 18–25
21. Knopke, I.: The importance of service and genre in recommendations for online radio and television programmes. In: Workshop on Music Recommendation and Discovery 2011, Workshop on Music Recommendation and Discovery (October 2011) 26–29
22. Popescu, G., Pu, P.: Probabilistic game theoretic algorithms for group recommender systems. In: Workshop on Music Recommendation and Discovery 2011, Workshop on Music Recommendation and Discovery (October 2011) 7–12
23. Ignatov, D., Poelmans, J., Zaharchuk, V.: Recommender System Based on Algorithm of Bicluster Analysis RecBi. In Ignatov, D., Poelmans, J., Kuznetsov, S., eds.: CEUR Workshop proceedings Vol-757, CDUD'11 - Concept Discovery in Unstructured Data. (2011) pp. 122–126
24. Ignatov, D.I., Kuznetsov, S.O.: Concept-based Recommendations for Internet Advertisement. In Belohlavek, R., Kuznetsov, S.O., eds.: Proc. CLA 2008. Volume Vol. 433 of CEUR WS., Palack University, Olomouc, 2008 (2008) 157–166
25. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1999)
26. Edwards, W., Barron, F.: SMARTS and SMARTER: Improved Simple Methods for Multiattribute Utility Measurement. Organizational Behavior and Human Decision Processes **60**(3) (1994) 306 – 325
27. Ignatov, D.I., Poelmans, J., Dedene, G., Viaene, S.: A New Cross-Validation Technique to Evaluate Quality of Recommender Systems. In Kundu, M.K., Mitra, S., Mazumdar, D., Pal, S.K., eds.: PerMIn. Volume 7143 of LNCS., Springer (2012) 195–202
28. Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. SIAM Rev. **51**(4) (November 2009) 661–703
29. Vander Wal, T.: Folksonomy Coinage and Definition (2007) http://vanderwal.net/folksonomy.html (accessed on 12.03.2012).
30. Ignatov, D.I., Kuznetsov, S.O., Magizov, R.A., Zhukov, L.E.: From Triconcepts to Triclusters. In: Proceedings of the 13th international conference on Rough sets, fuzzy sets, data mining and granular computing. RSFDGrC'11, Berlin, Heidelberg, Springer-Verlag (2011) 257–264