NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Dmitry I. Ignatov, Sergey I. Nikolenko,
Taimuraz Abaev, Jonas Poelmans

IMPROVING QUALITY OF SERVICE FOR RADIO STATION HOSTING: AN ONLINE RECOMMENDER SYSTEM BASED ON INFORMATION FUSION

BASIC RESEARCH PROGRAM

WORKING PAPERS

SERIES: MANAGEMENT
WP BRP 31/MAN/2014

Dmitry I. Ignatov, Sergey I. Nikolenko, Taimuraz Abaev, Jonas Poelmans

IMPROVING QUALITY OF SERVICE FOR RADIO STATION HOSTING: AN ONLINE RECOMMENDER SYSTEM BASED ON INFORMATION FUSION[1]

We present a new recommender system developed for the Russian interactive radio network *FMhost*. The system aims to improve the quality of this service; it is designed specifically to deal with small datasets, overcoming the shortage of data on observed user behavior. The underlying model combines a collaborative user-based approach with information from tags of listened tracks in order to match user and radio station profiles. It follows an adaptive online learning strategy based on both user history and implicit feedback. We compare the proposed algorithms with industry standard methods based on Singular Value Decomposition (SVD) in terms of precision, recall, and Normalized Discounted Cumulative Gain (NDCG) measures; experiments show that in our case the fusion-based approach produces the best results.

JEL Classification: M13, M15, C6, C7, C8

Keywords: e-commerce, quality of service, consumer behaviour, music recommender systems, interactive radio network, hybrid recommender system, information fusion, CRM

[1]This paper was prepared as a part of the project No 84-2014 Socio-political processes on the Internet: The structure and content of social interactions of the Basic Research Program, National Research University Higher School of Economics.

1 Introduction

The proliferation of online content often makes it difficult for content consumers to make choices. Many commercial websites, in their attempts to retain or broaden their user base, employ recommender systems to help their customers find goods, services, or content that would suit them most. The benefits of recommender systems are twofold: they not only facilitate the search for new items but also allow companies to sell additional goods or services that a user might miss otherwise. Recommender systems thus have been attracting more and more attention of researchers and practitioners as tools to improve service quality and to boost sales by recommending products based on previous consumer behavior or other features [1, 2, 3]. It has been shown that, for purposes of better customer relationship management, recommender systems can both increase user satisfaction and loyalty and, on the other hand, can give an insight for decision makers on the users' needs and behaviour and thus prevent user churn, improve user retention, and generally improve the service being provided [4].

A recommender system takes histories of user preferences together with additional information (demographic information about the users, information about the content of items being recommended etc.) and automatically creates personalized recommendations for goods and services. At present, the role of recommender systems in the world economy steadily increases; many modern markets, especially markets related to online purchases, are dominated by recommender systems. Offline stores also rely increasingly on recommender systems to improve product placement and guide customers through their shopping experience [5]. Recent economic studies indicate that recommender systems have important influence on the markets as they reduce product uncertainty for the customers, and recommendations can be viewed as information externalities [6, 7].

Another important application of recommender systems is the impact they provide on business management. Studies indicate that the presence of recommender systems significantly changes user behaviour, influencing both purchasing behaviour and trust [8, 9, 10]. Some recommender systems are based on web mining to train for consumer behaviour and adapt to it, providing important new prospects for e-commerce [11]; others provide important information for decision makers in marketing, benefiting the entire process of gathering and processing relevant information, making inferences from available information, and helping make better decisions for marketing campaigns [12]. Another important aspect is that recommender systems may change their properties with time, as new items and users accumulate in the system, so it is vitally important to have good monitoring tools that can pass information further to business management; this is an intersection of recommender systems and business intelligence [13].

Music recommendation is an important direction in the field of recommender systems. Many recent works in this area have appeared at the International Society for Music Information Retrieval Conference (ISMIR) [14], the Workshop on Music Recommendation and Discovery (WOMRAD) [15, 16], and the Recommender Systems conference (RecSys) [17]. Several broadcasting services, e.g., *last.fm*, *Yahoo!LaunchCast*, and *Pandora*, are known for their recommender systems and work on a commercial basis (however, the latter two do not operate in Russia). Despite many high quality works on different aspects of music recommendation, there are only a few studies devoted to online radio station recommender systems [18]. This work deals with the Russian online radio hosting service *FMhost*, in particular, with its new hybrid recommender system.

Recently, the focus of computer science research dealing with music has shifted from music information retrieval and exploration [19, 20, 21] to music recommendation [22, 23]. It is not a new direction (see, e.g., [24]); however, it is now inspired by new capabilities of large online services that can provide not only millions of tracks for listening, but also thousands of radio stations to choose from at a single web site. In addition to this, social tagging is an important factor that leads to new recommender algorithms based on tag similarity [25, 26, 27].

In this work, we consider the music recommendation problem from a slightly different angle. First, we recommend radio stations, i.e., sequences or sets of compositions rather than individual tracks, as most other music recommenders do. Second, as we demonstrate below, the *FMhost* service does not have enough data for reasonable SVD-based recommendations; still, recommendations have to be provided. We propose a novel recommender algorithm that combines three data matrices: radio station visits, listenings to specific tags, and frequency of tags inside radio stations. We show experimental results on the *FMhost* dataset and propose a fusion of two different algorithms that can be tuned for specific quality metrics.

The paper is organized as follows. In Section 2, we briefly survey related work in music recommendation. Section 3 outlines the online radio service *FMhost*. In section 4, we propose a novel recommender model, two basic recommender algorithms, a third algorithm that combines them, and show the recommender system architecture. Quality of service (QoS) measurements, a comparison with an SVD-based approach, and some insights into *FMhost* user behaviour are discussed in Section 5. Section 6 concludes the paper.

2 Related work

Music recommendation has become especially important because modern systems that provide music to their users strive to help users find their own hidden gems in the "long tail" of the popularity distribution of musical compositions and/or collections of compositions such as, in our case, radio stations. Most music recommender systems operate under the general principles of collaborative filtering [28]. For instance, *last.fm* mines user tastes both explicitly, from "like" marks with which users express their interest in compositions, and implicitly, from compositions that the users actually listen to. Many music recommenders also incorporate content-based features, analyzing the sound files themselves. An interesting example of the latter is the *Pandora* service [29, 30]; *Pandora* decomposes a music composition into the so-called music genome: the "genes" of a composition are different for different music genres, and compositions are graded for various "genes" by professional musicologists. Still, even content-based systems usually employ collaborative filtering and use content features to supplement classical recommender algorithms, so basic ideas of collaborative filtering such as matrix factorization [31, 32, 33, 34, 35] and nearest neighbors in both user-based and item-based collaborative filtering [36, 37, 38, 39] certainly apply; see also recent surveys on recommender systems [40, 41, 42, 43].

A widely acclaimed public contest on music recommender algorithms, KDD Cup[2], was recently held by *Yahoo!*. In KDD Cup, track 1 was devoted to learning to predict users' ratings of musical items (tracks, albums, artists, and genres) where the items formed a taxonomy: tracks belong to albums, albums belong to artists, where albums and tracks are also tagged with genres. Track 2 aimed at developing learning algorithms for separating music tracks scored highly by specific users from tracks that have not been scored by them. It attracted a lot of attention to problems that are both typical for recommender systems and specific for music recommendation: scalability issues, capturing the dynamics and taxonomical properties of recommended items, and other problems [44]. Another major music recommender contest with open data, the Million Songs Dataset Challenge[3], was held in 2012 by the Computer Audition Lab at UC San Diego and LabROSA at Columbia University [45]. The core data consists of triples $(user, song, count)$; it covers approximately 1.2 million users and more than 380,000 songs. While music recommender problems seldom have explicit user preferences (it is unlikely that users are prepared to rate every track they listen to), this kind of data can serve as input to one-class recommender algorithms that operate on implicit information regarding user activity [46, 47, 48]; these algorithms can also benefit from additional information about both users and items [49, 50].

Recent music recommendation trends reflect the advantages of hybrid approaches and call for user-centric quality measures [51]. For instance, the work [52] proposes a novel approach based on the so-called "forgetting curve" to evaluate "freshness" of predictions. Tags (both moderated and user-generated) turn out to be especially important for recommendations. The work [53] studies the problem of how much metadata one needs in music recommendation: a subjective evaluation of 19 users has revealed that pure content-based methods can be drastically improved with genre tags. Finally, the authors proposed a recommender approach that starts from an explicit set of music tracks provided by the user as evidence of his/her preferences and then infers high-level semantic descriptors for each track [54]. There is also a rich body of work devoted to tag suggestion and automated tag generation for music compositions since tags, especially generic tags like genres or social tags like "music for jogging" can then serve as preconstructed recommenders for the users [55, 56, 57, 58].

In [59], the authors proposed a music recommender system *Starnet* for social networking. It generates social recommendations based on positive ratings of friends, non-social recommendations based on positive ratings of other users in the network, and random recommendations. *Hottabs* is another interesting recently developed online music recommendation system [60], dedicated to learning how to play guitar. Some authors improve music recommender systems with semantic extraction techniques [61, 62]. In [63], the author describes a system of genre recommendation for music and TV

[2] `http://kddcup.yahoo.com/`

[3] `http://labrosa.ee.columbia.edu/millionsong/`

programs that can be considered as an alternative channel selector. The authors of [64] proposed a recommender system *GroupFan* which is able to aggregate preferences of a group of users to their mutual satisfaction.

Many online services (e.g., *last.fm* and *LaunchCast*) call their audio streams "radio stations". However, they are actually just playlists from a database of tracks which are based on a recommender system rather than real predefined channels. *FMhost*[4], on the other hand, provides users with online radio stations in the classical meaning of this term: human DJs perform live, and a radio station actually represents a strategy or mood of a specific person (DJ) who plays his/her own tracks, performs contests etc. Thus, the problem we aim to solve differs from most of the work done in terms of music recommendation, and some of the challenges are unique. For instance, our system is scalable and it allows users to listen to music without being registered, and as a result our test dataset contains only 4266 registered users with a recorded listening history. The dataset contains 2206 radio stations with 24803 non-zero entries in the user–station matrix; it is small and sparse, so standard recommendation techniques (e.g., SVD that we compare our algorithms to) fail to achieve good prediction quality.

3 Online service FMhost.me

3.1 A concise online broadcasting dictionary

We begin by briefly introducing some basic domain terminology. A *chart* is a track rating of a particular radio station; for example, a rock chart shows a certain number (e.g., 10) of most popular rock tracks, ranked from the most popular (rank 1) to the least popular (rank 10) according to a survey. A *live performance* (or just *live* for short) is a performance with one or several *DJs* (*disk jockeys*) assigned to it. They perform using their own PCs, and the audio stream is being redirected from them to an Icecast server and then distributed to the users. The DJs may also have their own blog for each live, where people can interact with DJs who perform live. *LiquidSoap* is a sound generator that broadcasts audio files (*.mp3, *.aac etc.) into an audio stream, and *Icecast* is a retranslation server that redirects an audio stream from one source (e.g., LiquidSoap) to many receivers.

3.2 The FMhost project

FMhost is an interactive radio network. The 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), and (4) radio station owner.

User capabilities vary with their status. Unauthorized listeners can listen to any station but cannot vote or become DJs and cannot use the recommender system and the rating system. Listeners can vote for tracks, lives, and radio stations. They are allowed to use the recommender system and the rating

[4] `http://host.fm/en/`

system, and 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, and (3) live performance. Stream redirection applies when a radio station owner has a separate dedicated server, uses *FMhost* as a broadcasting platform, and broadcasts with 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/she can download tracks, and then LiquidSoap generates the audio stream and the Icecast (`http://www.icecast.org/`) server redirects it to the listeners. Usually the owner sets a schedule for his radio station. Live performances are done by DJs. Everyone who has performed live at least once can be called a DJ. He or she 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 or she is in a crew.

FMhost was the first project of its kind in Russia, starting in 2009. Following *FMhost*'s success, there now exist several radio broadcasting portals: `http://frodio.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 major redesign of both codebase and recommender system architecture. In this paper, we describe the results of this upgrade.

The previous version of the recommender system experienced several problems, including tag discrepancy and personal tracks without tags at all. A survey conducted by *FMhost* with about a hundred respondents showed that more than half of them appreciated the previous version of its recommender system, and more than 80% of the answers were either positive or neutral (see [65, Table 1]); nevertheless, the new recommender model and algorithms provide even better recommendations and make even less prediction mistakes.

3.3 *FMhost* conceptual improvements

The new version features a more complex system of user interactions. 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. A new broadcasting panel allows the DJs to play tracks with new features 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 also be launched in the new version. 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 new system fixes all problems that were identified in the previous version.

4 Models, algorithms and recommender architecture

4.1 Input data and general structure

Our model is based on three data matrices [5]. 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. Elements of the second matrix $B = (b_{rt})$ show 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 has visited 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: $v^A = \sum_{t \in T} a_{ut}$, $v^B = \sum_{t \in T} b_{rt}$, and $v^C = \sum_{r \in R} a_{ur}$. We also introduce for each matrix A, B, C the corresponding frequency matrices A_f, B_f, and C_f; a frequency matrix results by normalizing the matrix with the respective vector of visits, e.g., $A_f = (a_{ut} \cdot (v^A_u)^{-1})$. Our model is not 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: Individual-Based Recommender System (IBRS) model, Collaborative-Based Recommender System (CBRS) model, and Fusion Recommender System (FRS) that aggregates the results of the first two. Each model has its own algorithmic implementation.

4.2 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 $(R_{u_0}, \preceq_{u_0}, \text{score})$, where R_{u_0} is a 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 score is a function which maps each radio station r from R_{u_o} to $[0,1]$.

The algorithm computes the l_1-norm (Manhattan) distance between a user u_0 and a radio station r: $d(u_0, r) = \sum t \in T |a_{u_0 t} - b_{rt}|$. Then distances between the user u_0 and all radio stations $r \in R$ are computed, and the algorithm constructs the relation $\prec_{u_0}$ according to the following rule: $r_i \preceq r_j$ iff $d(u_0, r_i) \le d(u_0, r_i)$. The function score operates on R_{u_0} according to the following rule:

$$\text{score}(r_i) = 1 - d(u_0, r_i) / \max_{r_j \in R} d(u_0, r_j). \tag{1}$$

Finally, after selecting N radio stations with N largest values in R_{u_0}, we have a ranked list $Top_N(u_0)$ of radio stations recommended to the user u_0. In case there are several elements with the same score (rank) so that $Top_N(u)$ is not uniquely defined, we simply choose the first elements according to some arbitrary ordering (e.g., lexicographically by their names).

[5] The data used for the experimenation is available by the following link: `http://bit.ly/hostfm`

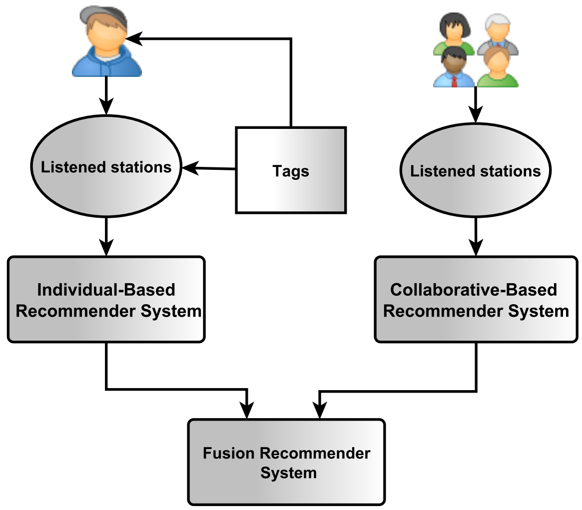

Figure 1: The recommender system architecture

As shown in Fig. 1, our simplified model takes into account only "listened tracks" but the previously proposed one [65] also deals with "liked tracks", "liked radio stations", and "favorite radio stations".

4.3 CBRS

The **CBRS** model is based on the C_f matrix (normalized number of times a user u has visited a radio station r). This matrix also yields a vector n^C which stores the total number of stations listened by each user $u \in U$. This vector also changes over time, and this value is used as a threshold to transform matrix C_f into a similarity matrix D via cosine similarity between users i and j:

$$sim(i,j) = \frac{c_{fi} \cdot c_{fj}}{\sqrt{\sum_{r \in R} c_{fir}^2 * \sum_{r \in R} c_{fjr}^2}} \tag{2}$$

After computing D, the algorithm constructs the list $Top_k(u_0) = (U_{u_0}, \preceq_{u_0}, \text{sim})$ of k users similar to the target user u_0 who awaits recommendations. We define the set of all radio stations user u_0 has listened to by $L(u_0) = \{r | c_{fur} = 0\}$. In a similar way, we define

$$\begin{aligned} Top_N(u_0) &= (R_{u_0}, \preceq_{u_0}, \text{score}), \\ \text{where } \text{score}(r) &= \text{sim}(u^*) \cdot c_{fu^*r} \\ \text{and } u^* &= \arg \max_{u \in U_{u_0}, r \in U/L(u_0)} \text{sim}(u) \cdot c_{fur}. \end{aligned} \tag{3}$$

Note that the variable score takes values in the range of $[0,1]$. The problem of choosing exactly the N best stations is solved in the same way as in the IBRS submodel.

4.4 FRS

After IBRS and CBRS have returned results, we dispose of two ranked lists of recommended stations $Top^I_N(u_0)$ and $Top^C_N(u_0)$ for a target user u_0 from IBRS and CBRS respectively. The **FRS** submodel proposes a simple solution for aggregating these lists into the final recommendation structure $Top^E_N(u_0) = (R^E_{u_0}, \preceq^E_{u_0}, \text{score}^E)$. For every $r \in R^C_{u_0} \cup R^I_{u_0}$, the function $\text{score}^E(r)$ maps r to the weighted sum

$$\beta \cdot \text{score}^C(r) + (1-\beta) \cdot \text{score}^I(r), \tag{4}$$

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

5 Quality of service assessment

To evaluate the quality of the developed system, we used a variant of the cross-validation technique proposed in [66]. We represent the dataset as an object-attribute table (binary relation) $T \subseteq U \times I$, where uTi iff user $u \in U$ used (purchased, watched, listened etc.) item $i \in I$. To evaluate the quality of recommendations in terms of precision and recall, we split the initial user set U into training and test subsets U_{train} and U_{test}, where the test set is smaller than the training set, e.g., with a $20/80$ proportion. Recommendation precision and recall are evaluated on the test set, and this part of the algorithm is similar to one step of conventional cross-validation. Then each user vector u from U_{test} is divided into two parts which consist of evaluated items I_{visible} and items I_{hidden} which we have intentionally hidden. Note that in existing literature, the proportion between the size of I_{visible} and I_{hidden} is not discussed even in similar schemes [67]. Then, for example, a user-based algorithm makes recommendations according to similarity between users from the test and training sets. Each user from U_{test} gets recommendations as a set of fixed size $r_n(u) = \{i_1, i_2, \cdots, i_n\}$. Precision and recall are defined as

$$\text{Recall} = \frac{|r_n(u) \cap u^I \cap I_{\text{hidden}}|}{|u^I \cap I_{\text{hidden}}|}, \tag{5}$$

$$\text{Precision} = \frac{|r_n(u) \cap u^I \cap I_{\text{hidden}}|}{|r_n(u) \cap I_{\text{hidden}}|}, \tag{6}$$

where u^I is the set of all items from I used by u. These measures are calculated for each user and then averaged. The experiment can be repeated several times, e.g. 100, for different test and training set splits, and then the values are averaged again. In addition, one can select the set I_{hidden} at random, but with a specific proportion, e.g. 20%. The idea behind the method comes from traditional cross-validation, but in the case of recommender systems some modifications are needed. We used a modified version of m-fold cross-validation, which is performed by splitting the initial set into m

disjoint subsets, where each subset is used as a test set and the other subsets are considered as training sets. To evaluate ranking quality based on the number of performed listenings we use the Normalized Discounted Cumulative Gain (NDCG) measure.

Before we proceed to the detailed description of the procedure, we discuss some important aspects of the *FMHost* data that we have mined.

5.1 Basic statistics

The dataset contains the following entities: users, tags, radio stations, and tracks. We have selected users with at least one tag in the profile, and then all the tags related to the selected users. At the next step, we have chosen radio stations that the selected users have listened to or with selected tags in the radio station profiles. Finally, we have chosen tracks related to at least one of the selected radio stations and to at least one tag. The resulting dataset contains 4266 users, 3618 tags, 2209 radio stations, and 4165 tracks. The corresponding matrices have the following number of nonempty entries: 38504 in the user–tag matrix, 18539 in the radiostation–tag, 24803 in the user–radiostation, 18781 in the track–tag, and 22525 in the radiostation–track matrix.

It is a well-known fact that social networking data often follows the so called power law distribution [68]. In order to choose which number of active users or radio stations we have to take into account for making recommendations, we performed a simple statistical analysis of the user and radio station activity. Around 20% of the users (only registered ones) were analysed.

Table 1: Basic parameters of the user and radio visits datasets, with 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**

Table 1 shows p-values of statistical tests performed with the *Matlab* package introduced in [68]. It shows 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 (which is actually a rather flat distribution compared to most services). Thus, if the service tends to take into account only active stations and users, it can cover 80% of all visits by considering 50% of their active audience. However, new radio stations still deserve to be recommended, so this

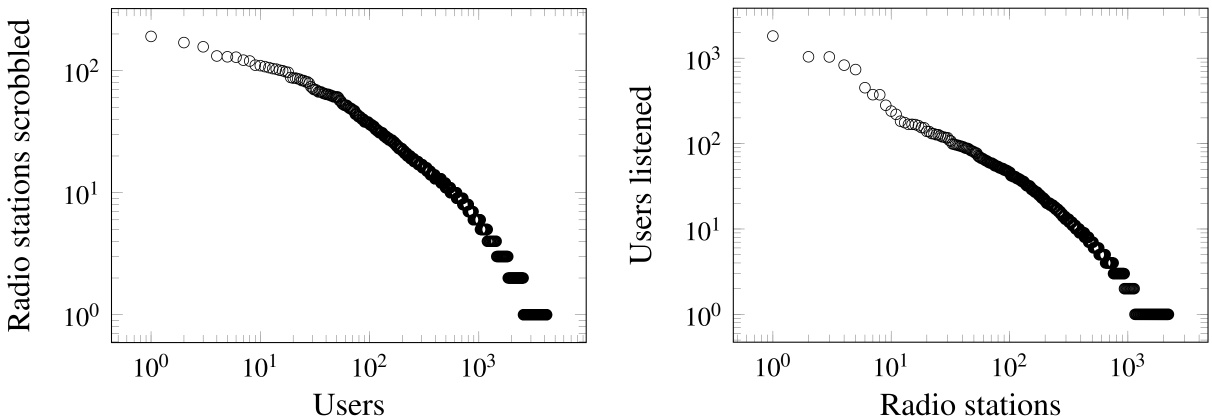

Figure 2: Power law at FMHost. On the left, users sorted by the number of radio stations they have listened to. On the right, radio stations sorted by the number of their users. Both graphs (on log scale) show power law dependencies.

rule can only be applied to the user database.

5.2 Quality assessment

To evaluate the quality of the IBRS subsystem, we compute average precision and recall on the set $R_N \subset R$, where N is the number of randomly "hidden" radio stations. We suppose that for every station $r \in R_N$ and every user $u \in U$ the algorithm does not know whether the radio stations were visited, and we change A_f and R accordingly. Then IBRS attempts to recommend the Top-N radio stations for the modified matrix A_f. Top-N average precision and recall are computed as follows:

$$\text{Precision} = \frac{\sum\limits_{u \in U} \frac{|R_u^I \cap L_u \cap R_N|}{|L_u \cap R_u^I|}}{|U|}, \tag{7}$$

$$\text{Recall} = \frac{\sum\limits_{u \in U} \frac{|R_u^I \cap L_u \cap R_N|}{|L_u \cap R_N|}}{|U|}. \tag{8}$$

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 the CBRS algorithm assuming that $c_{fu'r}$ is unchanged for $u' \in U/u$ and then compute

$$\text{Precision} = \frac{\sum\limits_{u \in U} \frac{|R_u^C \cap L_u \cap R_N|}{|L_u \cap R_u^C|}}{|U|}, \tag{9}$$

$$\text{Recall} = \frac{\sum\limits_{u \in U} \frac{|R_u^C \cap L_u \cap R_N|}{|L_u \cap R_N|}}{|U|}. \tag{10}$$

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

$$\beta^* = \arg\max_{\beta} \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{(\text{Precision} + \text{Recall})}. \tag{11}$$

After launching the system, we suppose to be able to collect enough reliable statistics in about one month of active operation in order to tune β and choose appropriate similarity and distance measures and 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-N precision and recall measures as well as NDCG. In addition, online surveys can be launched to assess user satisfaction with the new recommender system.

5.3 Experimental results

The IBRS algorithm uses the user–tag matrix A and the radiostation–tag matrix B. As a result, we have a matrix of predicted scores that contains recommendations of radiostations for users. Figure 6 shows the precision, recall, and NDCG measure of IBRS versus the size of the recommendation list. For the first recommended item, precision is about 30% and then it rapidly drops as Top-N grows to 5-10 elements. Then it becomes close to 1% while N goes to 100. Recall for the first recommended radiostation is 50%, and then the recall value slowly increases as Top-N grows. NDCG slowly increases while Top-N grows. Our *Matlab* implementation of IBRS runs for about 80s for all users.

The CBRS algorithm uses the user–radiostation matrix C. To evaluate its quality, we used 3×3-fold cross-validation which we described in the beginning of Section 5. Therefore, as a result we have 9 different partitions of the initial data into training and test set, and all measures are averaged by all partitions. Figure 6 shows that CBRS precision is smaller than IBRS precision for small Top-N size: for the first recommended radiostation it is about 7%. However, for large Top-N size it goes down to 2% ($N = 100$), which is better than IBRS precision. In terms of recall CBRS also loses: recall of the first recommended radiostation is about 10%, and then for higher values of N it becomes closer to IBRS recall. NDCG of CBRS is strictly less than the NDCG of IBRS, which shows that IBRS is a better ranker than CBRS. The testing time for the entire cross-validation procedure is about 50 minutes.

The hybrid recommender system FRS uses output matrices with scores of recommended radiostations for IBRS and CBRS. To make the final ranking, it employs a weighted sum of the IBRS and CBRS output matrices. Parameter β is tuned for each Top-N size. To this end, for β from 0 to 1 with step 0.05 we have calculated the resulting weighted matrix; for this matrix, we calculate precision, recall, the F-measure, and NDCG. The procedure is repeated for N ranging from 1 to 100. The value of β is then tuned to maximize one of the measures. By maximizing the F-measure, we improve the

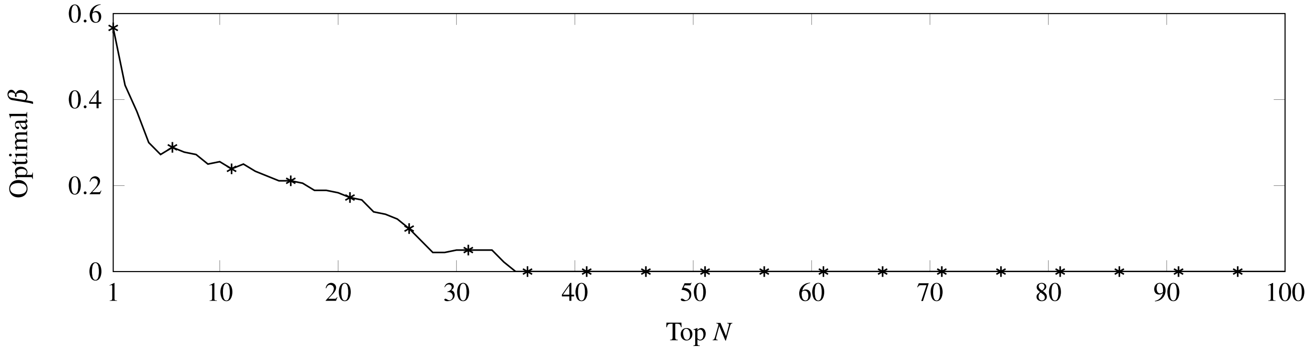

Figure 3: Tuning the parameter β to maximize F-measure

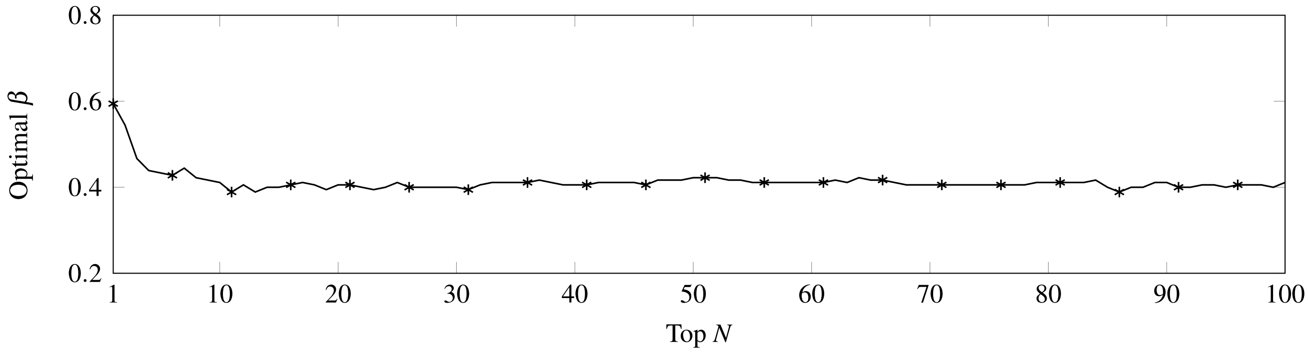

Figure 4: Tuning the parameter β to maximize NDCG

general quality of the output but nearly ignore the ranking. Maximization of NDCG takes into account the ranking. The time needed to tune β in the range from 0 to 1 with step 0.05 in our implementation is 200s. It takes up to 3s to calculate the final recommendation matrix with the chosen β.

When we maximize the F-measure with respect to β, for $N = 1$ IBRS has a slightly higher weight, about 0.57, but as Top-N grows to 30-35 radio stations, the parameter β smoothly decreases, thus increasing CBRS's contribution; for $N > 35$ β drops to 0 (see Fig. 3). These results can be explained by the fact that IBRS provides higher precision and recall than CBRS, but as N grows the F-meausre of CBRS is getting higher than for IBRS. As a result, the F-measure for FRS is not less than the F-measure of either IBRS or CBRS individually, and FRS performs efficiently for every tested recommender list size. In particular, for N in the order of 15-20 radiostations the F-measure of FRS is 2-3% higher than the F-measure of the best basic method (IBRS). For a larger size of the recommendation list, N, the FRS F-measure is close to CBRS.

NDCG maximisation by the parameter β prefers IBRS at first: the parameter begins with 0.6. But then in the range of N from 2 to 10 radiostations β decreases to 0.4 and stabilizes, oscillating for larger values of N around 0.40-0.41. This implies that CBRS contributes slightly more to the final recommendation (see Fig. 4). When we tune β to maximize NDCG, the weighted sum approach performed better than IBRS (which is better than CBRS with respect to NDCG) by 4-5% for all tested

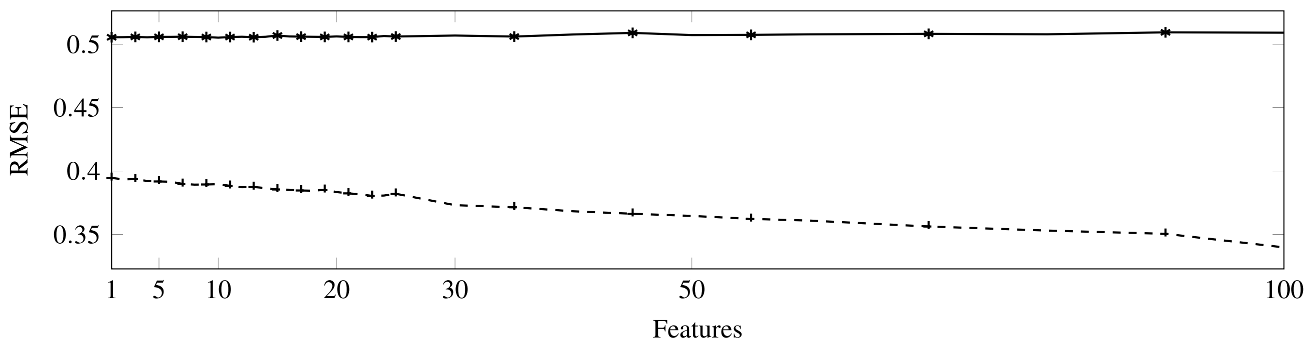

Figure 5: Root mean squared error in SVD experiments as a function of the number of features. Solid line denotes error on the validation set; dashed line, error on the training set.

output sizes.

For comparison, we have also implemented a standard collaborative filtering model based on Singular Value Decomposition (SVD) [69, 70, 32, 31, 28]. In this model, an item's rating is approximated as a scalar product of user and item feature vectors (plus some baseline predictors). In our setting, lacking explicit ratings, we have applied SVD to the matrix of user listenings to radio stations. Due to the power law dependencies we have discovered (see Section 5.1), we have used the logarithm of the number of listenings (plus one) in the model:

$$\log(\mathrm{listen}_{u,r}+1) \sim \mu + b_u + b_r + v_u^\top v_r, \tag{12}$$

where $\mathrm{listen}_{u,r}$ is the number of times user u listened to radio station r, μ is the general mean, b_u and b_r are the baseline predictors for the user u and the station r respectively, and v_u and v_r are the vectors of the user and station features respectively.

It was clear that there is not enough data and the matrix is too sparse for SVD. This was also supported by our experiments: we did not see any improvement at all as the number of SVD features grew, the quality on the validation set was stable all the way from a single feature to about 100 and then started showing clear signs of overfitting; this is depicted in Fig. 5. Results of the main experiments also supported this observation: in Fig. 6, SVD clearly loses to the methods proposed in this work.

6 Conclusion and further work

In this work, we have described the underlying models, algorithms, and system architecture of the new improved *FMHost* service and tested it on the available real dataset. We hope that the developed algorithms will help users find relevant radio stations for listening, and thus will provide FMHost with a more efficient tool to attract and retain customers. During future optimization and tuning of the algorithm, special attention should be paid to scalability issues and user-centric quality assessment.

By using bimodal cross-validation, we have built a hybrid algorithm FRS tuned to maximize either

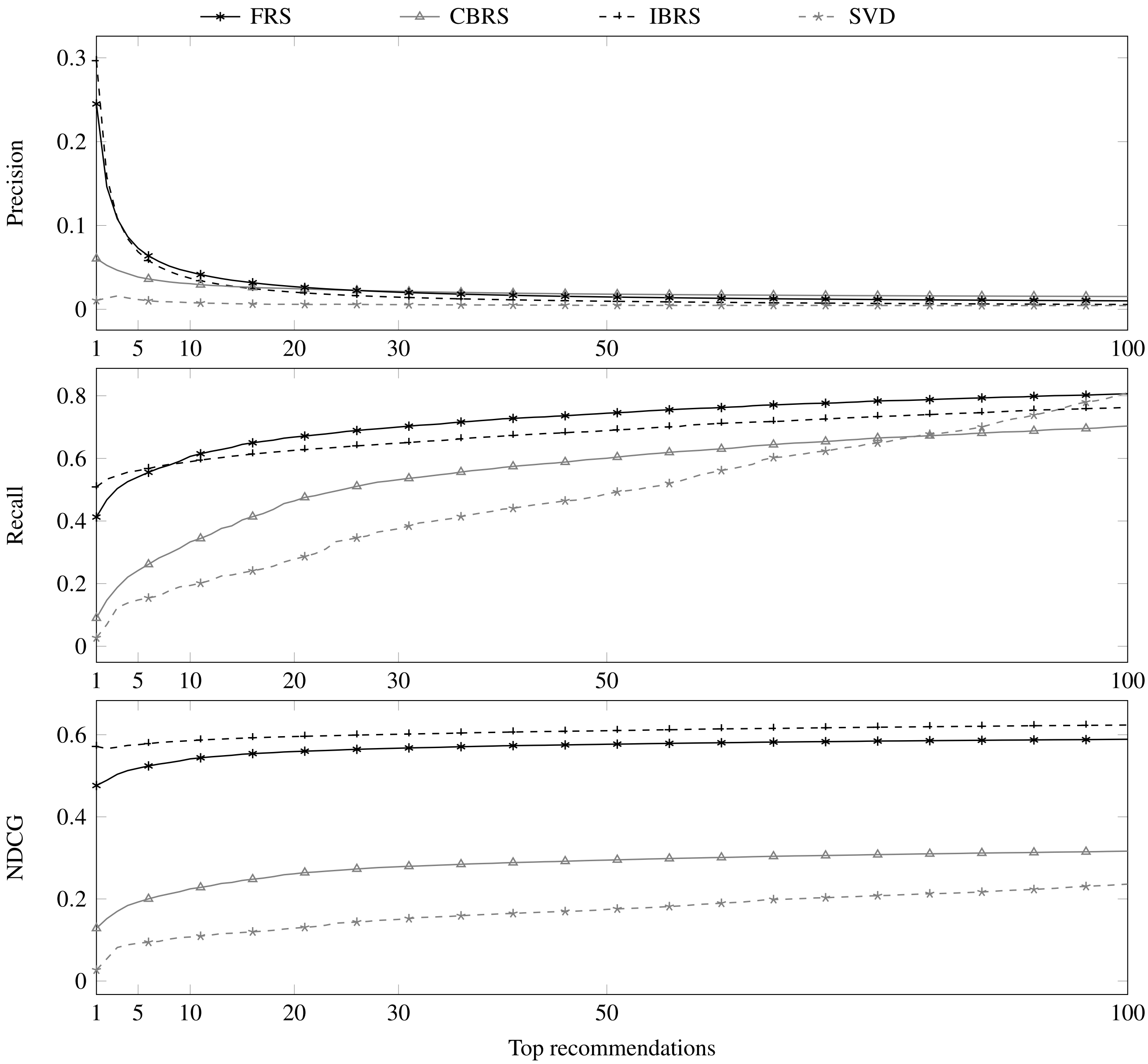

Figure 6: Experimental results for four algorithms: IBRS, CBRS, FRS, and SVD. Graphs, top to bottom: precision, recall, and NDCG as a function of the number of top recommendations predicted for a user (averaged over all users).

F-measure or NDCG for various values of N and β. The FRS algorithm performs better than the three other approaches, namely IBRS, CBRS, and SVD, both in terms of F-measure and in terms of NDCG.

According to the NDCG@n (NDCG at n) measure, IBRS is strictly better than CBRS, so the former one is a better ranker. The maximal value of F-measure is obtained for the Top-N of size 15.

Surprisingly, in our experiments the state-of-the-art SVD-based technique performed poorly in comparison to our proposed algorithms. This can be explained by the small size and sparseness of our dataset. We hypothesize that the methods described in this work will suit datasets with similar properties. Matrix factorization techniques remain an important tool to increase scalability, but they have to be carefully adapted and assessed, taking into account the folksonomic nature of the track tags and rather disappointing results of the SVD-based technique. Another important issue is connected to the triadic relational nature of the data (users, radio stations or tracks, and tags), which constitutes the so called *folksonomy* [71], a fundamental data structure in resource-sharing systems with tags. As shown in [72], 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.

It is worth noting that the proposed recommender system is not used for recommendation of radio streamers [73] but users receive recommendations on real radiostations headed by DJs. Its IBRS component may have an advantage on usage of the proposed adaptive user and radio profiles represented by tags' frequencies since both the user's tastes and radio repertoire may evolve over time. In this case pure collaborative-based approach like CBRS may become cumbersome to timely response to the user's needs.

Acknowledgments

We would like to thank Rimma Ahmetsafina, Mykola Pechenizkiy and Rustam Tagiew for their comments and Vasily Zaharchuk, Natalia Konstantinova and Andrey Konstantiov for their very important work at the previous stages of this project. Last but not least, we are thankful to Olessia Koltsova, the head of LINIS lab, for her support and help during the paper preparations. Our research was done in the framework of the Basic Research Program at the National Research University Higher School of Economics in 2014, at the Laboratory of Intelligent Systems and Structural Analysis (Moscow) and Laboratory of Internet Studies (St. Petersburg). Dmitry Ignatov was supported by Russian Foundation for Basic Research (grant # 13-07-00504).

References

[1] A. Fong, B. Zhou, S. Hui, G. Hong, T. A. Do, Web content recommender system based on consumer behavior modeling, Consumer Electronics, IEEE Transactions on 57 (2) (2011) 962–969. doi:10.1109/TCE.2011.5955246.

[2] B. Kim, S. Kim, A new recommender system to combine content-based and collaborative filtering systems, Journal of Database Marketing & Customer Strategy Management 8 (3) (2001) 244–252.

[3] A. Geyer-Schulz, M. Hahsler, M. Jahn, myvu: A next generation recommender system based on observed consumer behavior and interactive evolutionary algorithms, in: W. Gaul, O. Opitz, M. Schader (Eds.), Data Analysis, Studies in Classification, Data Analysis, and Knowledge Organization, Springer Berlin Heidelberg, 2000, pp. 447–457. doi:10.1007/

978-3-642-58250-9_36.
URL http://dx.doi.org/10.1007/978-3-642-58250-9_36

[4] Y.-F. Wang, D.-A. Chiang, M.-H. Hsu, C.-J. Lin, I.-L. Lin, A recommender system to avoid customer churn: A case study, Expert Systems with Applications 36 (4) (2009) 8071 – 8075. doi:http://dx.doi.org/10.1016/j.eswa.2008.10.089.
URL http://www.sciencedirect.com/science/article/pii/S0957417408008725

[5] D. Buser, Context-based recommender systems in conventional grocery–an economic analysis, in: System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, 2007, pp. 168b–168b. doi:10.1109/HICSS.2007.147.

[6] D. Bergemann, D. Ozmen, Optimal pricing with recommender systems, in: Proceedings of the 7th ACM Conference on Electronic Commerce, EC '06, ACM, New York, NY, USA, 2006, pp. 43–51. doi:10.1145/1134707.1134713.
URL http://doi.acm.org/10.1145/1134707.1134713

[7] F. Harper, X. Li, Y. Chen, J. Konstan, An economic model of user rating in an online recommender system, in: L. Ardissono, P. Brna, A. Mitrovic (Eds.), User Modeling 2005, Vol. 3538 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 307–316. doi:10.1007/11527886_40.
URL http://dx.doi.org/10.1007/11527886_40

[8] M. Gorgoglione, U. Panniello, A. Tuzhilin, The effect of context-aware recommendations on customer purchasing behavior and trust, in: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11, ACM, New York, NY, USA, 2011, pp. 85–92. doi:10.1145/2043932.2043951.
URL http://doi.acm.org/10.1145/2043932.2043951

[9] H. Chen, The impact of comments and recommendation system on online shopper buying behaviour, Journal of Networks 7 (2) (2012) 345–350.

[10] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, C. Newell, Explaining the user experience of recommender systems, User Modeling and User-Adapted Interaction 22 (4-5) (2012) 441–504. doi:10.1007/s11257-011-9118-4.
URL http://dx.doi.org/10.1007/s11257-011-9118-4

[11] Z. Zeng, An intelligent e-commerce recommender system based on web mining, International Journal of Business and Management 4 (7) (2009) 10–14. doi:10.5539/ijbm.v4n7p10.

[12] L. Mocean, C. M. Pop, Marketing recommender systems: A new approach in digital economy, Informatica Economica 16 (4) (2012) 142–149.

[13] C. Flix, C. Soares, A. Jorge, J. Vinagre, Monitoring recommender systems: A business intelligence approach, in: B. Murgante, S. Misra, A. Rocha, C. Torre, J. Rocha, M. Falco, D. Taniar, B. Apduhan, O. Gervasi (Eds.), Computational Science and Its Applications ICCSA 2014, Vol. 8584 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 277–288. doi:10.1007/978-3-319-09153-2_21.
URL http://dx.doi.org/10.1007/978-3-319-09153-2_21

[14] A. de Souza Britto Jr., F. Gouyon, S. Dixon (Eds.), Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013, Curitiba, Brazil, November 4-8, 2013, 2013.

[15] A. Anglade, C. Baccigalupo, N. Casagrande, Ò. Celma, P. Lamere, Workshop report: Womrad 2010, in: X. Amatriain, M. Torrens, P. Resnick, M. Zanker (Eds.), RecSys, ACM, 2010, pp. 381–382.

[16] A. Anglade, O. Celma, B. Fields, P. Lamere, B. McFee, Womrad: 2nd workshop on music recommendation and discovery, in: Proceedings of the fifth ACM conference on Recommender systems, RecSys '11, ACM, New York, NY, USA, 2011, pp. 381–382. doi:10.1145/2043932.2044013.
URL http://doi.acm.org/10.1145/2043932.2044013

[17] Q. Yang, I. King, Q. Li, P. Pu, G. Karypis (Eds.), Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013, ACM, 2013.

[18] N. Aizenberg, Y. Koren, O. Somekh, Build your own music recommender by modeling internet radio streams, in: A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, S. Staab (Eds.), WWW, ACM, 2012, pp. 1–10.

[19] O. Hilliges, P. Holzer, R. Klüber, A. Butz, Audioradar: A metaphorical visualization for the navigation of large music collections, in: A. Butz, B. D. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics, Vol. 4073 of Lecture Notes in Computer Science, Springer, 2006, pp. 82–92.

[20] D. F. Gleich, M. Rasmussen, K. Lang, L. Zhukov, The world of music: User ratings; spectral and spherical embeddings; map projections, Online report (2006).

[21] D. F. Gleich, L. Zhukov, M. Rasmussen, K. Lang, The World of Music: SDP Embedding of High Dimensional data, in: Information Visualization 2005, 2005, interactive Poster.

[22] K. Brandenburg, C. Dittmar, M. Gruhne, J. Abeer, H. Lukashevich, P. Dunker, D. Grtner, K. Wolter, H. Grossmann, Music search and recommendation, in: B. Furht (Ed.), Handbook of Multimedia for Digital Entertainment and Arts, Springer US, 2009, pp. 349–384.

[23] Ò. Celma, Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space, Springer, 2010.

[24] P. Avesani, P. Massa, M. Nori, A. Susi, Collaborative radio community, in: Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH '02, Springer-Verlag, London, UK, UK, 2002, pp. 462–465.
URL `http://dl.acm.org/citation.cfm?id=647458.728391`

[25] P. Symeonidis, M. M. Ruxanda, A. Nanopoulos, Y. Manolopoulos, Ternary semantic analysis of social tags for personalized music recommendation, in: J. P. Bello, E. Chew, D. Turnbull (Eds.), ISMIR, 2008, pp. 219–224.

[26] A. Nanopoulos, D. Rafailidis, P. Symeonidis, Y. Manolopoulos, Musicbox: Personalized music recommendation based on cubic analysis of social tags, IEEE Transactions on Audio, Speech & Language Processing 18 (2) (2010) 407–412.

[27] Y.-H. Yang, D. Bogdanov, P. Herrera, M. Sordo, Music retagging using label propagation and robust principal component analysis, in: Mille et al. [74], pp. 869–876.

[28] Y. Koren, R. M. Bell, Advances in collaborative filtering, in: F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011, pp. 145–186.

[29] M. Castelluccio, The music genome project, Strategic Finance 88 (6) (2006) 57–58.

[30] J. Joyce, Pandora and the music genome project, Scientific Computing 23 (10) (2006) 40–41.

[31] Y. Koren, R. M. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, IEEE Computer 42 (8) (2009) 30–37.

[32] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, IEEE Computer Society, Las Vegas, Nevada, USA, 2008, pp. 426–434.

[33] H. Ma, H. Yang, I. King, M. R. Lyu, Semi-nonnegative matrix factorization with global statistical consistency in collaborative filtering, in: Proceedings of the 18th ACM International Conference on Information and Knowledge Management, 2009, pp. 767–776.

[34] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: Advances in Neural Information Processing Systems, Vol. 20, 2008.

[35] R. Salakhutdinov, A. Mnih, Bayesian probabilistic matrix factorization using Markov chain Monte Carlo, in: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 880–887.

[36] Y. Koren, Factor in the neighbors: Scalable and accurate collaborative filtering, ACM Transactions on Knowledge Discovery from Data 4 (1).

[37] A. Said, B. J. Jain, B. Kille, S. Albayrak, Increasing diversity through furthest neighbor-based recommendation, in: Proceedings of the WSDM'12 Workshop on Diversity in Document Retrieval (DDR'12), 2012.

[38] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, 2001, pp. 285–295.

[39] P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom, J. T. Riedl, Grouplens: An open architecture for collaborative filtering of netnews, in: 1994 ACM Conference on Computer Supported Collaborative Work Conference, Association of Computing Machinery, Association of Computing Machinery, Chapel Hill, NC, 1994, pp. 175–186. doi:http://doi.acm.org/10.1145/192844.192905.

[40] L. L, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, T. Zhou, Recommender systems, Physics Reports 519 (1) (2012) 1 – 49, recommender Systems. doi:http://dx.doi.org/10.1016/j.physrep.2012.02.006.
URL http://www.sciencedirect.com/science/article/pii/S0370157312000828

[41] J. Bobadilla, F. Ortega, A. Hernando, A. Gutirrez, Recommender systems survey, Knowledge-Based Systems 46 (0) (2013) 109 – 132. doi:http://dx.doi.org/10.1016/j.knosys.2013.03.012.
URL http://www.sciencedirect.com/science/article/pii/S0950705113001044

[42] J. A. Konstan, J. Riedl, Recommender systems: from algorithms to user experience, User Modeling and User-Adapted Interaction 22 (1-2) (2012) 101–123. doi:10.1007/s11257-011-9112-x.
URL http://dx.doi.org/10.1007/s11257-011-9112-x

[43] X. Zhou, Y. Xu, Y. Li, A. Josang, C. Cox, The state-of-the-art in personalized recommender systems for social networking, Artificial Intelligence Review 37 (2) (2012) 119–132. doi:10.1007/s10462-011-9222-1.
URL http://dx.doi.org/10.1007/s10462-011-9222-1

[44] N. Koenigstein, G. Dror, Y. Koren, Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy, in: Proceedings of the fifth ACM conference on Recommender systems, RecSys '11, ACM, New York, NY, USA, 2011, pp. 165–172. doi:10.1145/2043932.2043964.
URL http://doi.acm.org/10.1145/2043932.2043964

[45] B. McFee, T. Bertin-Mahieux, D. P. W. Ellis, G. R. G. Lanckriet, The million song dataset challenge, in: Mille et al. [74], pp. 909–916.

[46] R. Pan, Y. Zhou, B. Cao, N. Liu, R. Lukose, M. Scholz, Q. Yang, One-class collaborative filtering, in: Proceedings of the 8th IEEE International Conference on Data Mining, 2008, pp. 502–511.

[47] R. Pan, M. Scholz, Mind the gaps: Weighting the unknown in large-scale one-class collaborative filtering, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, ACM, New York, NY, USA, 2009, pp. 667–676. doi:10.1145/1557019.1557094.
URL http://doi.acm.org/10.1145/1557019.1557094

[48] V. Sindhwani, S. S. Bucak, J. Hu, A. Mojsilovic, One-class matrix completion with low-density factorizations, in: Proceedings of the 10th IEEE International Conference on Data Mining, ICDM '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 1055–1060. doi:10.1109/ICDM.2010.164.
URL http://dx.doi.org/10.1109/ICDM.2010.164

[49] Y. Li, J. Hu, C. Zhai, Y. Chen, Improving one-class collaborative filtering by incorporating rich user information, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010, pp. 959–968.

[50] T. Yuan, J. Cheng, X. Zhang, Q. Liu, H. Lu, A weighted one class collaborative filtering with content topic features, in: S. Li, A. El Saddik, M. Wang, T. Mei, N. Sebe, S. Yan, R. Hong, C. Gurrin (Eds.), Advances in Multimedia Modeling, Vol. 7733 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 417–427. doi:10.1007/978-3-642-35728-2_40.
URL http://dx.doi.org/10.1007/978-3-642-35728-2_40

[51] O. Celma, P. Lamere, Music recommendation and discovery revisited, in: Proceedings of the fifth ACM conference on Recommender systems, RecSys '11, ACM, New York, NY, USA, 2011, pp. 7–8. doi:10.1145/2043932.2043936.
URL http://doi.acm.org/10.1145/2043932.2043936

[52] Y. Hu, M. Ogihara, Nextone player: A music recommendation system based on user behavior, in: Klapuri and Leider [75], pp. 103–108.

[53] D. Bogdanov, P. Herrera, How much metadata do we need in music recommendation? a subjective evaluation using preference sets, in: Klapuri and Leider [75], pp. 97–102.

[54] D. Bogdanov, M. Haro, F. Fuhrmann, A. Xambó, E. Gómez, P. Herrera, Semantic audio content-based music recommendation and visualization based on user preference examples, Inf. Process. Manage. 49 (1) (2013) 13–33.

[55] D. Eck, P. Lamere, T. Bertin-Mahieux, S. Green, Automatic generation of social tags for music recommendation., in: J. C. Platt, D. Koller, Y. Singer, S. T. Roweis (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc., 2007.
URL `http://dblp.uni-trier.de/db/conf/nips/nips2007.html#EckLBG07`

[56] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, B. Kégl, Aggregate features and AdaBoost for music classification, Machine Learning 65 (2-3) (2006) 473–484. `doi:10.1007/s10994-006-9019-7`.
URL `http://dx.doi.org/10.1007/s10994-006-9019-7`

[57] Z. Zhao, X. Wang, Q. Xiang, A. M. Sarroff, Z. Li, Y. Wang, Large-scale music tag recommendation with explicit multiple attributes, in: Proceedings of the International Conference on Multimedia, MM '10, ACM, New York, NY, USA, 2010, pp. 401–410. `doi:10.1145/1873951.1874006`.
URL `http://doi.acm.org/10.1145/1873951.1874006`

[58] F. Font, J. Serr, X. Serra, Folksonomy-based tag recommendation for online audio clip sharing., in: F. Gouyon, P. Herrera, L. G. Martins, M. Mller (Eds.), Proceedings of the 13th International Society for Music Information Retrieval Conference, FEUP Edies, 2012, pp. 73–78.
URL `http://dblp.uni-trier.de/db/conf/ismir/ismir2012.html#FontSS12`

[59] C. S. Mesnage, A. Rafiq, S. Dixon, R. P. Brixtel, Music discovery with social networks, in: Workshop on Music Recommendation and Discovery 2011, Workshop on Music Recommendation and Discovery, 2011, pp. 1–6.
URL `http://ceur-ws.org/Vol-793/womrad2011_complete.pdf`

[60] M. Barthet, A. Anglade, G. Fazekas, R. Kolozali, Sefki Macrae, Music recommendation for music learning: Hotttabs, a multimedia guitar tutor, in: Workshop on Music Recommendation and Discovery 2011, Workshop on Music Recommendation and Discovery, 2011, pp. 7–13.
URL `http://ceur-ws.org/Vol-793/womrad2011_complete.pdf`

[61] I. Tatl, A. Birturk, Using semantic relations in context-based music recommendations, in: Workshop on Music Recommendation and Discovery 2011, Workshop on Music Recommendation and Discovery, 2011, pp. 14–17.
URL `http://ceur-ws.org/Vol-793/womrad2011_complete.pdf`

[62] P. Knees, M. Schedl, 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, 2011, pp. 18–25.
URL http://ceur-ws.org/Vol-793/womrad2011_complete.pdf

[63] I. Knopke, 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, 2011, pp. 26–29.
URL http://ceur-ws.org/Vol-793/womrad2011_complete.pdf

[64] G. Popescu, P. Pu, Probabilistic game theoretic algorithms for group recommender systems, in: Workshop on Music Recommendation and Discovery 2011, Workshop on Music Recommendation and Discovery, 2011, pp. 7–12.
URL http://ceur-ws.org/Vol-793/womrad2011_complete.pdf

[65] D. I. Ignatov, A. V. Konstantinov, S. I. Nikolenko, J. Poelmans, V. Zaharchuk, Online recommender system for radio station hosting, in: N. Aseeva, E. Babkin, O. Kozyrev (Eds.), BIR, Vol. 128 of Lecture Notes in Business Information Processing, Springer, 2012, pp. 1–12.

[66] D. I. Ignatov, J. Poelmans, G. Dedene, S. Viaene, A New Cross-Validation Technique to Evaluate Quality of Recommender Systems, in: M. K. Kundu, S. Mitra, D. Mazumdar, S. K. Pal (Eds.), PerMIn, Vol. 7143 of LNCS, Springer, 2012, pp. 195–202.

[67] P. Symeonidis, A. Nanopoulos, A. N. Papadopoulos, Y. Manolopoulos, Nearest-biclusters collaborative filtering based on constant and coherent values, Inf. Retr. 11 (1) (2008) 51–75.

[68] A. Clauset, C. R. Shalizi, M. E. J. Newman, Power-law distributions in empirical data, SIAM Rev. 51 (4) (2009) 661–703. doi:10.1137/070710111.

[69] R. M. Bell, Y. Koren, Scalable collaborative filtering with jointly derived neighborhood interpolation weights, in: Proceedings of the 7$^{\text{th}}$ IEEE International Conference on Data Mining, IEEE Computer Society, Omaha, Nebraska, USA, 2007, pp. 43–52.

[70] R. M. Bell, Y. Koren, Lessons from the Netflix Prize challenge, SIGKDD Explorations 9 (2) (2007) 75–79.

[71] T. Vander Wal, Folksonomy Coinage and Definition, http://vanderwal.net/folksonomy.html (accessed on 12.03.2012) (2007).
URL http://vanderwal.net/folksonomy.html

[72] D. I. Ignatov, S. O. Kuznetsov, J. Poelmans, L. E. Zhukov, Can triconcepts become triclusters?, Int. J. General Systems 42 (6) (2013) 572–593.

[73] O. Moling, L. Baltrunas, F. Ricci, Optimal radio channel recommendations with explicit and implicit feedback, in: P. Cunningham, N. J. Hurley, I. Guy, S. S. Anand (Eds.), Sixth ACM

Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012, ACM, 2012, pp. 75–82. doi:10.1145/2365952.2365971.
URL http://doi.acm.org/10.1145/2365952.2365971

[74] A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, S. Staab (Eds.), Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume), ACM, 2012.

[75] A. Klapuri, C. Leider (Eds.), Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011, University of Miami, 2011.

Authors' details

Dmitry I. Ignatov
Department of Data Analysis and Artificial Intelligence, Computer Science Faculty and Laboratory of Intelligent Systems and Structural Informatics,
National Research University Higher School of Economics, Russia

Sergey I. Nikolenko
Laboratory of Internet Studies, National Research University Higher School of Economics, ul. Soyuza Pechatnikov, d. 16, 190008 St. Petersburg, Russia, sergey@logic.pdmi.ras.ru
Steklov Mathematical Institute, nab. r. Fontanka, 27, 191023 St. Petersburg, Russia

Taimuraz Abaev
Alumni of National Research University Higher School of Economics, Russia

Jonas Poelmans
Department of Data Analysis and Artificial Intelligence, Computer Science Faculty,
National Research University Higher School of Economics, Russia