COLLABORATIVE FILTERING RECOMMENDER SYSTEMS IN MUSIC RECOMMENDATION

Urszula Kużelewska[1], Rafał Ducki[2]

[1] Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

[2] Student of Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: Nowadays, the primary place of information exchange is the internet. Its features, such as: availability, unlimited capacity and diversity of information influenced its unrivalled popularity, making the internet a powerful platform for storage, dissemination and retrieval of information. On the other hand, the internet data are highly dynamic and unstructured. As a result, the internet users face the problem of data overload. Recommender systems help the users to find the products, services or information they are looking for.
The article presents a recommender system for music artist recommendation. It is composed of user-based as well as item-based procedures, which can be selected dynamically during a user's session. This also includes different similarity measures. The following measures are used to assess the recommendations and adapt the appropriate procedure: RMSE, MAE, Precision and Recall. Finally, the generated recommendations and calculated similarities among artists are compared with the results from LastFM service.

Keywords: collaborative filtering, music recommendations, recommender systems

1. Introduction

Recommender systems (RS) are methods approaching the problem of information filtering. Their task is to register and analyse a user's preferences and generate a personalised list of items. In other words, the systems filter the information that may be presented to the user based on their interest. As input data, they register products' ratings, views of Web sites, purchases of items, as well as specific characteristics or descriptions of the products [11].

Recommendation concerns, among the others, news, music, video, content of e-learning courses, books and subject of web sites or web site navigation.

Advances in Computer Science Research, vol. 10, pp. 67-79, 2013.

Music is regarded as particularly difficult domain for recommender systems application [3]. It combines the fields of music information retrieval (MIR) and recommendations [14]. There are several approaches addressed this problem. The easiest solution is to gather ratings from users, however this type of data is difficult to obtain and can contain, sometimes intended, outliers and noise. The other approach is to count tracks played by users and process them to form ratings, e.g. LastFM (http://www.lastfm.com) . Finally, input data can be users' playlists composed of their favourite songs and artists. There are also methods, which process music streams extracting fundamental complex features from the records, e.g. Mufin (http://www.mufin.com), Pandora (http://www.pandora.com).

The article presents a recommender system for music artist recommendation. It uses track play counts as input data. Different RS approaches, including similarity measures, have been implemented and evaluated using efficiency coefficients and compared to LastFM service results. The paper is organised as follows: the next section introduces recommender system domain: classification, problems, similarity measures and evaluation. The following part presents selected music recommendation solutions. The last two sections concern experiments as well as analysis of the results and the final conclusions.

2. Introduction to recommender systems

Recommender systems help customers to find interesting and valuable resources in the internet services. Their priority is to create and examine users individual profiles, which contain their preferences, then update the service content to finally increase the user's satisfaction. This section introduces recommender systems: their classification and main problems. It presents selected similarity measures and lists the most common approaches to recommendations evaluation.

2.1 Classification and problems in recommender systems

Considering a type of input data as well used methods, recommendation systems are divided into content-based, collaborative filtering (CF), knowledge-based and hybrid [9].

Content-based recommendations (called content-based filtering) base on attribute (characteristic) vectors of items created from text connected with the items, e.g. their description, genre, etc [11]. As an example, in case of books, the item characteristics include its genre, topic or author. The content-based algorithms recommend items, which are similar to highly rated by the user other items in past. As an example, if a user liked (rated or bought) X movie, a recommender system searched other movies, which were similar to X with regard to its genre, title, director's name or description of the story. The main advantages of content-based systems are: relatively simple implementation and independence of users. The disadvantages are: a problem of "cold start" for users and the requirement of items' features analysis.

Knowledge-based approach is better for one-time users stores, e.g. selling cameras (people do not buy cameras often) [1]. The approach bases on technical attributes of the items and user preferences, also weighted, related to the attributes. Knowledge acquirement is often realised by interaction with users. This is an approach, where the "cold start" problem does not appear and users' data are not required to store for long time, however they have to use specific techniques to gather the knowledge.

Collaborative filtering techniques search similarities among users or items, however only archives of users behaviour are analysed [1]. As an example, similar users have mostly the same products in their baskets and similar items are bought by the same customers. This is the most preferred technique in recommender systems. They based on the assumption, that if two users have the same opinion on the particular item, it is very likely they like similarly other items. The most important advantages of this kind of systems are: high precision, simple implementation, no additional knowledge about a domain or objects. The long list of advantages is supplemented with the following disadvantages: a problem of "cold start" for users and objects and poor scalability.

Hybrid approach combines at least two different methods: problems in each of them are solved by strengths of the other one.

The most often issue in RS domain is cold-start problem [9]. It concerns a new user case, when there is no information about their preferences, and a new item, when a new object is added to the offer. Due to the fact, that the new object is not assigned to any user, it can't be recommended to anyone. Content-based approach solves this issue by calculating similarity between the new and already stored items basing on their features.

In arbitrary recommender system application, the number of offered items is large, whereas a user during one session visits a few to tens of them. It results in sparsity of input data and lower reliability in terms of measuring the similarity between customers [4].

Finally, however vitally important challenge in the field of on-line recommendations is scalability. RS deal with large amount of dynamic data, however the time of results generation should be reasonable to apply them in real-time applications. A user reading news expects to see next proposition for him/her in seconds, whereas millions of archived news have to be analysed [4].

2.2 Methods of similarity calculation in collaborative filtering systems

The final content of a recommendation list significantly depends on the similarity measure chosen for the recommendation system. To measure closeness between points $x = [x_1, x_2, \ldots, x_m]$ and $y = [y_1, y_2, \ldots, y_m]$ one can use measures, which have been adapted from relationship or distance among objects coefficients or coefficients created especially for recommendations. For all similarity measures, their higher values indicate higher degree of similarity.

The most popular measure is cosine based one, which calculates cosine of the angle between objects (see Equation 1). When two users or items are similar they have comparable ratings, therefore they are close in space and have the same direction from the origin. The value for the closest points is equal 1, whereas for the farthest: -1.

$$s_{cosine}(x,y) = \frac{\sum_{i=1}^{m} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{m} x_i^2} \cdot \sqrt{\sum_{i=1}^{m} y_i^2}} \tag{1}$$

Another example of similarity measure is Pearson correlation, which calculates the tendency of two series of paired values to move together proportionally. The correlation is described by Equation 2 and has value from the interval [-1,1].

$$s_{Pearson}(x,y) = \frac{\sum_{i=1}^{m} (x_i - \overline{x}) \cdot (y_i - \overline{y})}{\sqrt{\sum_{i=1}^{m} (x_i - \overline{x})^2} \cdot \sqrt{\sum_{i=1}^{m} (y_i - \overline{y})^2}} \tag{2}$$

Pearson correlation, although simple and often used in early research papers, suffers from several disadvantages. First of all, the value does not consider relationship between overlap values and the size of vectors. The other one is an undefined value if a user has the same preference for all items.

Another variant of correlation based similarity is Spearman's rank correlation coefficient. It also measures the relation between variables, however instead of the preference values their relative ranks are taken for computation. The ranks are based on order of ratings, therefore the lowest score of a user have a rank equal 1, the following one - a rank equal 2, etc. Equation 3 describes the coefficient; the rank vectors x' and y' correspond to the preferences vectors, respectively x and y.

$$s_{Spearman}(x,y) = 1 - \frac{6 \cdot \sum_{i=1}^{m} (x'_i - y'_i)}{m \cdot (m^2 - 1)} \tag{3}$$

Tanimoto coefficient is a measure, which ignores preference values taking into account sets of objects appearing in both vectors (see Equation 4).

$$s_{Tanimoto}(x,y) = \frac{|x \cap y|}{|x \cup y|} \tag{4}$$

Distance based measures are commonly used for measure similarity among objects. To adapt their values for increasing according to their closeness rising one can use Equation 5, in which $d(x,y)$ denotes distance between x and y.

$$s(x,y) = \frac{1}{1 - d(x,y)} \tag{5}$$

Distance based similarity's values are from interval (0,1]. The most popular distance measures in recommender systems are Euclidean and Manhattan metrics.

2.3 Evaluation of recommender systems

Evaluating recommender systems and their algorithms is a difficult task [8]. The main reason is their data dependent performance. The approach should be adjusted to different values of ratings as well as to predominant number of users or items. The second reason is changing data in time. Users' tastes vary in time as new products or items appear. Finally, many recommender systems use different metrics as well as new measures are proposed, which are also dependent on a specific dataset.

However, there are measures, which are often used to assess prediction accuracy. Predictive accuracy metrics measure how close the recommender system's predicted ratings are to the true user ratings [8].

The most common example is RMSE, which has been popularised in Netflix prize (www.netflixprize.com). It is described by Equation 6, in which the difference between predicted p and real r rating is calculated. Similar measure is MAE (see Equation 7), however it is more tolerant for high rating divergences. Both metrics should have the lowest values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{m}(p_i - r_i)^2} \tag{6}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{m}|p_i - r_i| \tag{7}$$

Another approach to recommendations evaluation uses metrics from information retrieval domain: *Precision* and *Recall*. They were applied for RS by Sarwar [12]. Items can appear in recommendation list and be relevant (N_{rr}) or irrelevant (N_{ri}). They can be not recommended, as well, but be in fact relevant (N_{nr}) or irrelevant

(N_{ni}). *Precision* is defined as the ratio of relevant items recommended to number of all items in the recommendation list (Equation 8), whereas *Recall* is defined as the ratio of recommended relevant items to total number of relevant items available (Equation 9).

$$Precision = \frac{N_{rr}}{N_{rr} + N_{ri}} \tag{8}$$

$$Recall = \frac{N_{rr}}{N_{rr} + N_{nr}} \tag{9}$$

Precision represents the probability that a recommended item is relevant, whereas *Recall* calculates the probability that a relevant item is recommended. It is desirable to have the highest values of these metrics.

3. Music recommendation

Music is a part of people's everyday life. We can listen to the music on the radio, on the internet or buy albums in stores. However, only promoted or the most popular music is easy to find. Recommender systems are a good tool to address the problem.

The most popular approaches to music recommendation are: collaborative filtering, content-based information retrieval, emotion-based and context-based models [14].

The collaborative filtering music recommenders base on history of track plays or direct music ratings. Interesting solution is automated playlist generation [3], where collocated artists are identified basing on their occurrence in the playlists.

Content based procedures analyse songs' description, features or acoustic characteristics [5]. Based on extracted features, data mining algorithms, such as clustering or kNN classification is applied.

Similar to content-based approach, the emotion-based models base on patterns of acoustic characteristic, however prefer perceptual features such as energy, rhythm, temporal, spectral, and harmony [2].

Context-based approach uses public opinion to discover and recommend music [10]. Popular social networks websites provide rich human knowledge such as comments, music reviews, tags and friendship relations. The context-based techniques collect the information to identify artist similarity, genre classification, emotion detection or semantic space.

Music database is an example of extremely huge size source of data. There is a large number of music artists, however there is far more of music fans. Although there are popular music discovery websites such as LastFM, Allmusic (http://www.allmusic.com) or Pandora (http://www.pandora.com), many new methods in scientific articles appear. The most often proposed are the hybrid recommender systems, which are a good solution to cope with the size of data [6].

4. Experiments

This section presents results of experiments with a hybrid system of music recommendation. The system was created as a part of master thesis in Bialystok University of Technology [7]. It is a Web application using Apache Mahout library [16].

Training data was extracted from LastFM music service in form of text files presented in Figure 1. The set contained ratings of 500 users, who listened 13680 tracks of 4436 artists. On average, one user listened to 27.36 songs, whereas one artist was played 3.08 times. Each row of the file contains a user id, date and time of the listening, an id and a name of the artist, an id and a name of the track played by the user.

user_000001	2009-05-02T15:24:45Z	3d05eb8b-1644-→	4Hero	cc7da9f6-df0e-4f88-a921-0f3b13f51fd1	Stoke Up The Fire
user_000001	2009-05-02T15:19:46Z	3d05eb8b-1644-→	4Hero	0024d72c-136f-49f2-9078-ce4b39b94d3f	Something In The W
user_000001	2009-05-02T15:13:49Z	3d05eb8b-1644-→	4Hero	2f550569-8859-4345-a554-ff698eef3ffe	Play With The Char
user_000001	2009-05-02T15:08:57Z	3d05eb8b-1644-→	4Hero	00b811a4-762b-492e-b5ef-9a673a55da57	Give In
user_000001	2009-05-02T15:05:06Z	3d05eb8b-1644-→	4Hero	b79e44f0-2a27-4f50-8a08-ce959a48c9c0	Sink Or Swim
user_000001	2009-05-02T15:00:59Z	3d05eb8b-1644-→	4Hero	6b71d43e-9258-4abd-89da-921fe00070a1	Look Inside
user_000001	2009-05-02T14:56:52Z	3d05eb8b-1644-→	4Hero	0fef630e-df3d-4880-97b6-06eb3e2767a7	Morning Child
user_000001	2009-05-02T14:56:52Z	3d05eb8b-1644-→	4Hero	8e9ec4d9-2248-4f6f-af74-2982f107159a	Take My Time
user_000001	2009-05-02T14:52:15Z	3d05eb8b-1644-→	4Hero	0fef630e-df3d-4880-97b6-06eb3e2767a7	Morning Child

Fig. 1. Data extracted from LastFM service

The aim of the practical part of the article was to construct and evaluate a recommender system in real environment (see Figure 2). It concerns the source of data, deployment of the application on the server as well as the way of evaluation, which was comparison to recommendation lists from LastFM. To make its recommendations effective, the system had to adapt the procedure of recommendations to an active user basing on implemented measures of errors.

One of the main problem was to preprocess the data assigning users to tracks with a rating value. It was performed using the count of track plays. The first step was to normalize the number of plays (see Equation 10) and rank the results with integers from the interval [1,2,3,4,5]. The symbols used in the equation are the following: $r(u_i,t_j)$ - means the rating value, $|u_i(t_j)|$ is a number of plays of track t_j of user u_i, $|u_i|$ is a total number of plays listened by the user u_i and the remaining component denotes maximal number of plays of a particular track listened by one of users from input data.

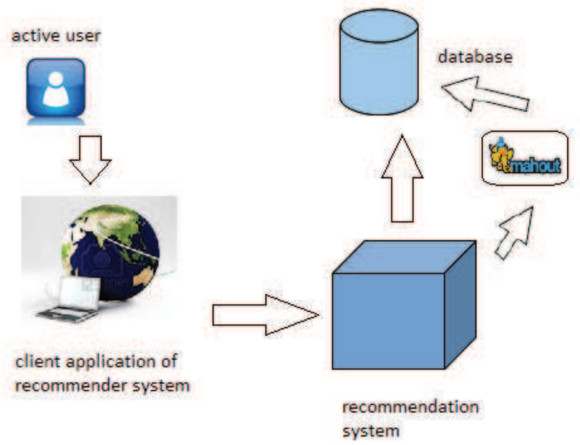

Fig. 2. The architecture of the created recommender system

$$r(u_i,t_j) = \frac{|u_i(t_j)|}{|u_i| \cdot \max_{x=1,y=1}^{x=m,y=n}\{u_x(t_y)\}} \tag{10}$$

The normalisation operation does not affects the data relationship; the graphs presenting the most often played artists and their popularity for randomly selected user are similar (see Figure 3).

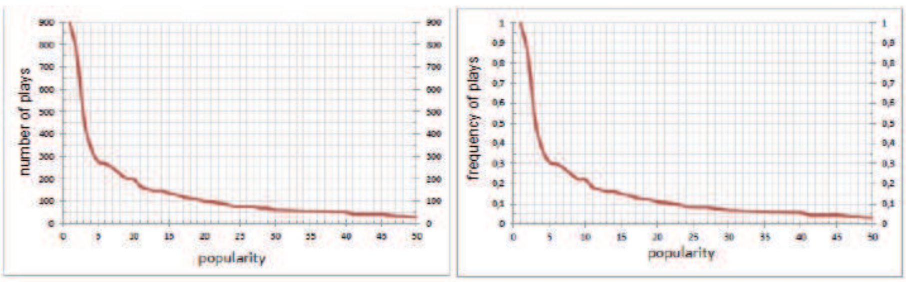

Fig. 3. The most often played 50 artists popularity for randomly selected user before (left) and after (right) data processing

Density of the prepared users' rating matrix calculated using Equation 11 (p is a number of ratings, m - a number of users and n - a number of artists) was 0.62%, which is high enough to apply collaborative filtering procedures and not influencing negatively on time of a generated list of recommendations.

$$\rho = \frac{p}{m+n} \tag{11}$$

The value 1 was the most frequent rating in the processed data (47.41%), followed by 2 (25.58%), 3 (11.24%), 5 (9.9%)and 4 (5.86%). It is worth mentioning that the choice of the ratings range was not only dictated by its popularity. The experiments results performed for the range [1,..,10] using RMSE value were worse.

The algorithms taken for the experiments were user-based as well as item-based collaborative filtering methods. The similarity measures were: cosine measure, Pearson and Spearman correlation, Tanimoto coefficient, Manhattan and Euclidean distance based measures.

In user-based approach it is necessary to determine neighbourhood of an active user. The most popular approach is to identify its k nearest neighbours (*kNN* method). A number of the neighbours is important and affects precision of recommendations. Table 1 contains the results of RMSE for various number of k.

Table 1. RMSE in user-based approach for various number of neighbours and different similarity measures.

k	Manhattan dist. based	Euclidean dist. based	Cosine similarity	Pearson correlation	Spearman rank correlation	Tanimoto coefficient
5	1.35	1.41	1.70	1.53	1.63	1.33
10	1.56	1.31	1.56	1.51	1.61	1.31
15	1.67	1.36	1.59	1.54	1.62	1.33
20	1.76	1.36	1.60	1.53	1.65	1.34
25	1.79	1.35	1.63	1.54	1.65	1.34
30	1.79	1.35	1.60	1.53	1.65	1.34
35	1.79	1.35	1.60	1.53	1.63	1.33
40	1.81	1.33	1.59	1.55	1.61	1.33
45	1.78	1.31	1.56	1.56	1.60	1.33
50	1.80	1.32	1.54	1.57	1.61	1.33
75	1.75	1.30	1.45	1.60	1.58	1.33
100	1.72	1.30	1.40	1.61	1.62	1.33
250	1.42	1.29	1.34	1.61	1.67	1.32
500	1.33	1.29	1.34	1.61	1.67	1.32

In most of the similarity measures cases the value of RMSE decreases when the size of neighbourhood rises. The exception are both correlation coefficients. The greater number of neighbour users requires more time to identify and process them. Taking the mentioned information into account, the optimal results is ~ 1.30 generated for 100 and 250 size of neighbourhood and Euclidean based as well as Tanimoto coefficients.

The best values of user-based results were compared to item-based approach (see Table 2). In this case the best similarity is Manhattan distance based measure. In case of Spearman correlation no results were generated. In all cases, the item-based solution generates recommendations much faster that user-based methods.

Table 2. Comparison of user-based and item-based approach for different similarity measures.

Similarity	user-based			item-based		
	MAE	RMSE	time [s]	MAE	RMSE	time [s]
Manhattan	0.97	1.35	0.704	0.65	0.99	0.007
Euclidean	0.96	1.30	0.303	0.82	1.15	0.009
Cosine	1.08	1.40	0.320	0.85	1.17	0.008
Pearson	1.13	1.53	0.302	1.06	1.51	0.010
Spearman	1.18	1.60	0.322	-	-	-
Tanimoto	0.98	1.31	0.312	0.84	1.16	0.005

The results presented above show, that there is no procedure of recommender system, which is able to be the most effective during its work. It depends on the ratio of the number of users and items as well as on density of the rating matrix. One of the solutions is to combine various approaches and to change the systems according to their performance. In the system presented in this paper the method of recommendation generation is changed in case of efficiency deterioration.

Finally, the system composed of the selected RS methods was compared with LastFM results. The highest coherency of recommendation lists was equal 35%. The similarity between artists was also compared as follows. For every singer or band from training data the system generated 100 most similar the other artists. The list was compared to 100 most similar artists from LastFM service generated for the same singer or band. Figure 4 presents the percentage measure values of coherence for each of the artists. One can notice similarity about 30-40%, however there are many values approximately equal 90%.

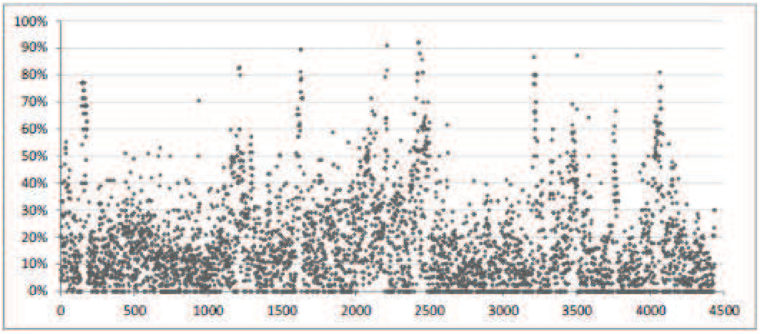

Fig. 4. Comparison of artists' similarity coherence calculated by the proposed system and LastFM service

5. Conclusions

This article presents an approach to music recommendation based on predictive efficiency, which was tested in real environment. The recommendations proposed in the created system are evaluated using error based coefficients as well as compared to real results from the LastFM service. The system combined different approaches: user-based and item-based to generate the recommendations effectively. Despite small size of training data, the obtained results were better than the authors expected. Thereby, they considered them as satisfactory.

Future research can be connected with attachment content based part for similarity calculation among artists as well as tracks. Description of artists or tags for tracks can be taken as input data. Due to limitations of predictive approach of evaluation, the other effectiveness measures can be used. An example is Intra-List Similarity Measure [15], which increases diversification of recommendations with regard to their sort or topic. Another approach is to track users' clickstream. If an active user selects none of the proposed tracks, the procedure of recommendation can be changed and a new list generated.

References

[1] S.S. Anand, B. Mobasher, Intelligent techniques for web personalization, Lecture Notes in Computer Science, vol.3169 (2005), pp. 1-36.

[2] K. Bischoff et all, Music mood and theme classification - a hybrid approach, 10th International Society for Music Information Retrieval Conference, 2009, pp. 657-662.

[3] G. Bonnin, D. Jannach, Evaluating the quality of playlists based on hand-crafted samples, 14th International Society for Music Information Retrieval Conference, 2013, pp. 263-268.

[4] D. Bridge, J. Kelleher, Experiments in Sparsity Reduction: Using Clustering in Collaborative Recommenders, Lecture Notes in Computer Science, vol. 2464 (2002), pp. 144-149.

[5] M.A. Casey et all, Content-based music information retrieval: current directions and future challenges, Proceedings of IEEE, vol.96, no. 4 (2008), pp. 668-696.

[6] O. Celma, Music Recommendation and discovery - the long tail, long fail, and long play in the digital music space, Springer, 2010

[7] R. Ducki, System rekomendujący wykonawców muzycznych (in Polish), Master Thesis supervised by U. Kużelewska, Faculty of Computer Science, Bialystok University of Technology, 2013.

[8] J.L. Herlocker, Evaluating Collaborative Filtering Recommender Systems, ACM Transactions on Information Systems, vol. 22, no. 1 (2004), pp. 5–53.

[9] D. Jannach et all, Recommender Systems: An Introduction, Cambridge University Press, 2010

[10] S. Panagiotis et all, Ternary Semantic Analysis of Social Tags for Personalized Music Recommendation, 9th International Society for Music Information Retrieval Conference, 2008, pp. 219-224.

[11] F. Ricci et all, Recommender Systems Handbook, Springer, 2010.

[12] B. Sarwar et all, Analysis of recommendation algorithms for E-commerce, 2nd ACM Conference on Electronic Commerce, 2000, pp. 285–295.

[13] B. Sarwar et all, Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering, 5th International Conference on Computer and Information Technology, 2002.

[14] Y. Song,S. Dixon, M. Pearce, A Survey of Music Recommendation Systems and Future Perspectives, 9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012), 2012, pp. 395-410.

[15] C.N. Ziegler et all, Improving Recommendation Lists through Topic Diversification, 14th International Conference on WWW, ACM Press, 2005, pp. 22-32.

[16] Apache Mahout, Open-source data mining library, [http://www.mahout.apache.org], accessed 20.10.2013.

SYSTEMY TYPU *COLLABORATIVE FILTERING* W REKOMENDACJI MUZYKI

Streszczenie W obecnych czasach głównym miejscem wymiany informacji jest internet. Jego cechy, takie jak: wysoka dostępność, nieograniczona pojemność i różnorodność informacji wpłynęły na jego niezrównaną popularność. W ten sposób internet stał się potężną platformą do przechowywania, rozpowszechniania i udostępniania informacji. Z drugiej strony, dane internetowe są bardzo dynamiczne i niestrukturalizowane. W rezultacie, użytkownicy internetu muszą radzić sobie z problemem przeładowania danych. Systemy rekomendujące służą pomocą użytkownikom w celu znalezienia poszukiwanych produktów, usług lub informacji.

W artykule przedstawiono system rekomendujący artystów muzycznych. Składa się on z procedur typu user-based oraz item-based oraz różnych sposobów szacowania podobieństwa, które mogą być zmieniane dynamicznie podczas sesji użytkownika. Do oceny list rekomendacji wykorzystano następujące miary: RMSE, MAE, Precision i Recall. Dodatkowo, wygenerowane rekomendacje i obliczone podobieństwa między artystami są porównywane z wynikami z serwisu LastFM .

Słowa kluczowe: systemy rekomendujące, rekomendacja muzyki, wspólna filtracja

Artykuł zrealizowano w ramach pracy badawczej S/WI/3/2013.