# Music Information Retrieval – Soft Computing Versus Statistics

Bozena Kostek(✉)

Faculty of Electronics, Telecommunications and Informatics, Audio Acoustics Laboratory,
Gdańsk University of Technology, Narutowicza 11/12 80-233, Gdańsk, Poland
bokostek@audioakustyka.org

**Abstract.** Music Information Retrieval (MIR) is an interdisciplinary research area that covers automated extraction of information from audio signals, music databases and services enabling the indexed information searching. In the early stages the primary focus of MIR was on music information through Query-by-Humming (QBH) applications, i.e. on identifying a piece of music by singing (singing/whistling), while more advanced implementations supporting Query-by-Example (QBE) searching resulted in names of audio tracks, song identification, etc. Both QBH and QBE required several steps, among others an optimized signal parametrization and the soft computing approach. Nowadays, MIR is associated with research based on the content analysis that is related to the retrieval of a musical style, genre or music referring to mood or emotions. Even though, this type of music retrieval called Query-by-Category still needs feature extraction and parametrization optimizing, but in this case search of global on-line music systems and services applications with their millions of users is based on statistical measures. The paper presents details concerning MIR background and answers a question concerning usage of soft computing versus statistics, namely: why and when each of them should be employed.

**Keywords:** Music information retrieval (MIR) · Feature extraction · Soft computing · Collaborative filtering (CF) · Similarity measures

## 1 Introduction

Music Information Retrieval is a very well-exploited field. They are venues devoted to MIR only (e.g. ISMIR, MIREX) [18][20][24][34] in which state-of-the-art MIR methods and achievements are critically evaluated, also sessions, workshops, discussion panels dedicated to this domain occur within artificial intelligence, audio, multimedia and other symposia and conferences [14][16][17][23]. On the other hand, there exist many music recommendation services, commercial and non-commercial that are based on social networking rather than on MIR-related methods [31][38][39[40][41]. This is often the case when a query for specific song or music genre may be performed based on similarity measures retrieved from large music archives [23][31]. In this context the stress should be on 'large' because smaller databases could easily be managed by human resources. Most prior research done into the audio genre recognition within the field of Music Information Retrieval were based on rather small music databases with a few classes of music genres [1][8][9][17][25].

K. Saeed and W. Homenda (Eds.): CISIM 2015, LNCS 9339, pp. 36–47, 2015.
DOI: 10.1007/978-3-319-24369-6_3

Even though collaborative (commercial) music services exist for many years, there are still some key problems that should be addressed in this field. This is especially important in cases when human-based evaluation doesn't always give correct answers or is far from giving correct answers, but nevertheless the user-based annotation is utilized in predicting the user's music preference. Moreover, there remains the key problem related to the scalability of the proposed solutions, regardless of the type of application (research- or commercial-based).

Applications that may be discerned within MIR area are: music genre classification, automatic track separation, music transcription, music recommendation, music generation, music emotion recognition (MER), music indexing, intellectual property rights management, and others. Many of the applications recalled above are based on a similar approach that consists in music pre-processing including parametrization, and the usage of soft computing methods [9][10][11][12][13][15][19][23][35]. These background notions are to be shortly reviewed with a focus on whether they need to be readdressed by MIR community.

The aim of this paper is paper is two-fold. First of it discusses MIR-related research and background measures utilized in music services. Secondly, it is to answer a question concerning usage of soft computing versus statistics, namely: why and when each of them should be employed. Section 2 reviews state-of-the-art in MIR in the context of search-based analysis, while in Section 3 issues related parametrization and soft-computing-based approach to genre classification are shown. Section 4 reviews background research that lies behind the song prediction in music services. Finally, summary remarks are formulated forming Conclusion Section.

## 2 Queries in MIR

Without any doubts one may say that MIR is a global research concentrated on the practical use of technical implementations and systems applications to music. Supported by soft computing, MIR evolved into a new domain, namely musical informatics. One of the crucial issues is the improvement of the efficiency of music recognition (e.g. in terms of performance), close to classification performed by human. However one of the problems is that human-based evaluation is not always able to give correct answers. Thus, we expect better soft-computing- than human-based performance. This concerns both music genre and emotion recognition.

During its early stages, MIR focus was on studies that allowed for searching for music information through QBH, *Query-by-Humming/Singing/Whistling*. Since singing or whistling is a natural ability of humans, humming to a microphone seemed to be the most convenient way to search for a given melody.

Full representation of a non-polyphonic piece is often called 'melody profile'. 'Sequence of frequencies' is a representation losing time-domain information, that is onsets and durations of sounds, but the information about pitches is preserved. In melodies represented as 'sequences of intervals', tonality information is lost, but the sizes of intervals between each pair of two consecutive sounds are known. The most simplified representation is the 'sequence of the directions of intervals' – only the directions of pitch of subsequent sounds in a melody are known. The last representation contains significantly reduced information, but at the same time preserves enough information for retrieval, i.e. it is resistant to rhythm changes (as no rhythm is retained), tonality and transposition errors. One of the most often cited work within QBH research studies is by Ghias *et al.* [5], who implemented a system able to detect coarse melodic contour based on Parsons code and retrieved by text string search. Even though the system had some constraints, i.e. usage of MIDI code, easily discerned notes, no rhythmic information, each pair of consecutive notes simply coded as “U” (“up”) if the second note is higher than the first note, “R” (“repeat”) if the pitches are equal, and “D” (“down”) otherwise, it performed surprisingly well on a prepared database. This is especially interesting, when one takes into account fact that a human ability to recognize hummed melody is not very high. It was observed that the average human accuracy in recognizing hummed queries is approximately 66% [27][37]. This is why formulating queries in that way may not be fully sufficient, even though it is intuitive for humans or musicians, but still may be inconvenient for non-musicians.

A simple measure used for non-polyphonic pieces and single-channel melodies, which are common in the MIDI standard is based on the distance *d* between a query $q=q_1, q_2, \ldots, q_m$ *and* object $t= t_1, t_2, \ldots, t_n$ is calculated with Eq. 1. The length of the query equals to *m*, the length of the object is *n*. The average difference in pitch between the query and an object in the database is calculated, the minimum average difference is acknowledged to be the distance between the query and the object – the shifts of *j* positions in a melody and transpositions of Δ semitones are committed to minimize the value of the distance.

$$d(q,t)=\frac{1}{m}\min_{j=1}^{n-m}(\min_{\Delta}\sum_{i=1}^{i=m}|\, q_i - t_{i+j} + \Delta\,|) \tag{1}$$

In melody retrieval systems, a query is usually a fragment of a full melody, so the matching is done in many locations of a melody, the pattern given in the query is matched against all objects in the database. In addition, queries do not exactly match the melodies in the database, so time-consuming approximate string matching techniques should be used. All those factors enlarge the computational complexity of the music retrieval task. Although optimizations to the classical approach by Baesa-Yates to approximate string matching were proposed, algorithms for melody retrieval may still be time-consuming, especially if the database contains large amount of musical files or/and rhythm information by detecting periodicity in time domain [21] or by analyzing note duration is added to music database.

More advanced MIR applications support *Query-by-example* searching. They are strongly rooted in collaborative music social networking when a given song may be used as a query for similar music. However more broaden retrieval refers to *content-based* analysis. In particular, search for similar musical styles, genres or mood/emotions of a musical piece is called *Query-by-category: musical style, genre, mood/emotion* (content-based) [11]. These types of information retrieval are visible in both research-based and music services, however the most significant difference between these two approaches lies in the size of music databases. Research-based databases contain a few hundred or the most a few thousands music excerpts, typically 30 (or less) second-long because of their copyright situation which should allow processing and presenting them to the public, while music services offer million of songs. These facts translate straightforward into the answer when and why soft computing or similarity-based approach can be applied.

The need for user-centric music recommendation created music services. Search may use tags contained in the ID3v.2 format (a query may consist of the song title, artist, genre, composer, album title, song length, lyrics, etc.). Music databases contain songs assigned to music genres, described by low level feature vectors or higher level descriptors, such as an instrument name, lyrics, etc., often annotated manually. Music services collect also interaction traces between the user and the song or between the users. A simple "interaction" means to play a song by the user and save it to the list of the so-called "favorite songs" This is the way of creating the user's profile. This information can be sent from the user's computer in the form of an application (e.g. scrobbler - an application installed by the service last.fm [39], which involves automatic transmission of metadata for all of the music tracks for the user's computer for analysis and the so-called collaborative filtering (CF) [4][6][22][30].

With regard to the effectiveness of music search, when low level-feature-based approach is used for small databases, the achieved effectiveness varies depending on a feature vector of used parameters and decision algorithms and is in the range of 60-90% [20][16][17][37]. It should be noted that efficiency is comparable to results obtained in the process of musical genre classification by human. In the case of databases based on tags IDv3.2 format, the accuracy of searching depends on the efficiency of the search algorithm to search the database (e.g., SQL), which means that a typographical error contained in the query within the well-known music databases such as FreeDB or GRACENOTE may cause a lack of response.

Annotating music manually requires a large number of "experts" with musical background, and is time-consuming. However, when performed by statistically significant number of people participating in the process, this starts to be to some extent reliable and effective method. This method is also called social tagging and takes the form in which key words describing a musical piece are added by users. Of course, it must be remembered that manual annotation can also be problematic in the context of various musical tastes and preferences, which can lead to a situation where the same track is assigned to different genres by individuals with diverse musical experience. That is why, it is often observed that users are not able to fully objectively assign a given musical piece to appropriate musical styles.

## 3 Parametrization and Decision Systems

Paramerization, a part of the pre-processing, aims at differentiating objects between different classes, recognizing unclassified objects (from unknown class) and determining whether an object is a member of a certain class. The underlying need to parametrize musical signals is their redundancy, thus a parametrization process results in the creation of feature vectors. Therefore, the decision process can be based on a set of parameters that are characteristic for e.g. musical style. Feature vectors containing time-, frequency or time-frequency-domains descriptors are often completed by adding statistical parameters.

As mentioned already retrieval that performs based on a low-level description of music depends on the quality of parametrization and the associated decision system.

Low-level descriptors are usually based on the MPEG 7 standard, Mel-frequency cepstral coefficients (MFCC's) or, finally, dedicated parameters suggested by researchers [2][7][14][15][19][24][25]. Within this approach, feature descriptors are assigned to a given music excerpt in order to perform automatic annotation of a given piece. Thus, the adequate selection of parameters, the algorithm optimization in terms of signal processing and data exploration techniques serve as key technologies that provide effective music tagging automatically.

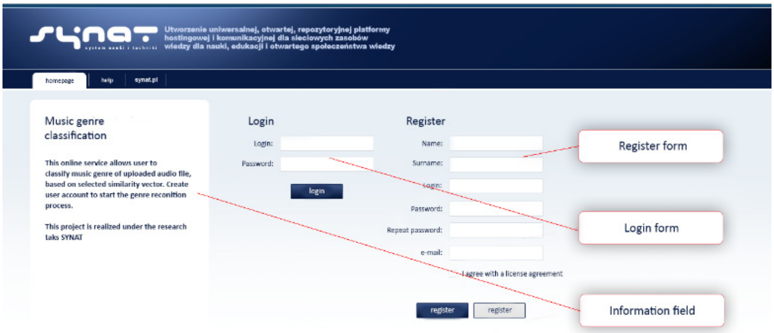
An example of a set of descriptors (191 in total) based on MPEG 7 standard, mel cepstral and dedicated parameters before optimization is given in Table 1 [7][10][28][29]. This was the feature vector created for the ISMIS'2011 (19th International Symposium on Methodologies for Intelligent Systems) music recognition contest. Prepared by the author and her collaborators was then incorporated into the ISMIS database [16]. The database contains over 1300 music excerpts, represented by 6 music genres (classical, Jazz, Blues, Rock, Heavy Metal, Pop). The winners of this competition got around 88% of correct classification [32]. As mentioned before the original FV contains 191 descriptors. Such a large number of parameters allows for an effective classification of musical genres, but at the same time it leads to a very high data redundancy, what results in a reduced classification effectiveness in terms of time consumption. That's why the author and her Ph.D. student applied PCA-based (Principal Component Analysis) optimization, and they obtained even higher accuracy in the classification process, but most important - classifying music genres was possible in real time based on buffered parts of the processed signals [7].

**Table 1.** Audio features an overview by the total number and description per type [7][29]

| No. of param. | Audio Feature Description |
|---|---|
| 1 | Temporal Centroid |
| 2 | Spectral Centroid and its variance |
| 34 | Audio Spectrum Envelope (ASE) in 34 subbands |
| 1 | ASE mean |
| 34 | ASE variance in 34 subbands |
| 1 | Mean ASE variance |
| 2 | Audio Spectrum Centroid (ASC) and its variance |
| 2 | Audio Spectrum Spread (ASS) and its variance |
| 24 | Spectral Flatness Measure (SFM) in 24 subbands |
| 1 | SFM mean |
| 24 | SFM variance |
| 1 | SFM variance of all subbands |
| 20 | Mel Cepstral Coefficients (MFCC) –first 20 |
| 20 | MFCC Variance –first 20 |
| 3 | No. of samples higher than single/double/triple RMS value |
| 3 | Mean of THR_[1,2,3]RMS_TOT for 10 time frames |
| 3 | Variance of THR_[1,2,3]RMS_TOT for 10 time frames |
| 1 | A ratio of peak to RMS (Root Mean Square) |
| 2 | A mean/variance of PEAK_RMS_TOT for 10 time frames |
| 1 | Number of transition by the level Zero |
| 2 | Mean/Variance value of ZCD for 10 time frames |
| 3 | Number of transitions by single/double/triple level RMS |
| 3 | Mean value of [1,2,3]RMS_TCD for 10 time frames |
| 3 | Variance value of [1,2,3]RMS_TCD for 10 time frames |

In general, the input signal is analyzed in the frequency sub-bands and then a set of parameters are calculated. That's why the optimization process of the feature vectors containing low-level features is further needed. Further, an important issue is that the available music excerpts represents typically 30 seconds (or less) of the whole track, which in most cases it is the beginning of the track (which not always is a perfect match for this music genre). Due to that fact even such a genre as Rap&HipHop which is quite easy to distinguish by the listener, can be hard to classify by the pre-trained system, since these 30 seconds can be represented either by the musical part or lyrics, differing much in style.

The same feature vector was applied to a larger database (but diminished to 173 parameters because not all frequency bands were present in the signal), called Synat, containing approximately 52,000 30 seconds-long music excerpts [7]. They are allocated to 22 music genres: Alternative Rock, Blues, Broadway&Vocalists, Children's Music, Christian&Gospel, Classic Rock, Classical, Country, Dance&DJ, Folk, Hard Rock&Metal, International, Jazz, Latin Music, Miscellaneous, New Age, Opera&Vocal, Pop, Rap&Hip-Hop, Rock, R&B, and Soundtracks. The database contains additional metadata, such as: artist name, album name, genre and song title. In addition to the items listed in the database, songs include also track number, year of recording and other parameters typically used for annotation of recordings. The user interface of this system is shown in Fig. 1.


Utworzenie uniwersalnej, otwartej, repozytoryjnej platformy hostingowej i komunikacyjnej dla sieciowych zasobów wiedzy dla nauki, edukacji i otwartego społeczeństwa wiedzy

Music genre classification

This online service allows user to classify music genre of uploaded audio file, based on selected similarity vector. Create user account to start the genre reconition process.

This project is realized under the research taks SYNAT

Login

Login:

Password:

Register

Name:

Surname:

Login:

Password:

Repeat password:

e-mail:

□ I agree with a license agreement

**Fig. 1.** Synat system user interface

From the whole Synat database 32,110 audio excerpts were chosen representing 11 genres (it is to note that this gives 5 555 030 parameters altogether, i.e. 32,110 x 173-element feature vector). That's why the PCA was applied to diminish this number for classification process. The system allows for an effective recommendation of music, experiments performed on 11 genres with an optimized feature vector returned classification accuracy of above 92% [11].

When reviewing MIR-related sources, one may see that among known classifiers the most often used are: SVMs (Support Vector Machines), minimum-distance methods, to which the *k*-Nearest Neighbor (*k*-NN) method belongs, Decision Trees, Random Forests, Rough Sets, etc. [15][17][26][34][35]. Each of these systems should ideally be considered in terms of high robustness and efficiency, not being computationally expensive, 'protecting' against overfitting, etc. Even though there is room to refine most of the given criteria, but when this list of criteria and conditions is reviewed one can say that the most problematic to achieve is the condition of not being computationally expensive when applied to large databases. That's why music recommendation services relies on statistics rather than on learning algorithms.

## 4 Music Recommendation

There are at least two layers of analysis when talking about music recommendation systems. It concerns both understanding and predicting user preferences. A recommender system must interact with the user, both to learn the user's preferences and provide recommendations based on the nearest neighbor for any query [4]. Systems should collect reliable data from which to compute their recommendations and preferences, reducing the noise in user preference data sets.

Before some background notions related to recommender systems are recalled, the problem of scalability should be pointed out, first. Scalability of search solutions imposes either small databases (utilized in research) and typically not showing results in real-time or utilizing techniques that reduce the number of users or items (or/and both) in search. One of the well-known solutions aimed to reduce the complexity and high dimensionality of database spaces is *Locality Sensitive Hashing* (LSH) belonging to randomized algorithms [33]. Its role is not to return exact answer but to guarantee a high probability that will bring in an answer close to the correct one. The algorithm builds a hash table, i.e., a data structure that allows for mapping between a symbol (i.e., a string) and a value. Then an arbitrary, pseudorandom function of the symbol that maps the symbol into an integer that indexes a table is calculated [33]. LSH is based on the idea that, if two points are close in a predefined space, then after the mapping operation these two points will remain close together [33].

The basis for a collaborative filtering is a collection of users' preferences for various (music) items (see Fig. 2 for explanation) [6][10][22][30][36]. Preference expressed by the user for an item is called a rating. The (user-item ) matrix **X** with dimensions $K \times M$, is composed of $K$ users, and $M$ songs. A single matrix element is described by $x_{k,m} = r$, which means that the $k$th user assigns the $r$ rate for the $m$th song. The matrix **X** may be decomposed into row vectors, representing each individual rating and may be treated as a separate prediction for the unknown rating [6][10][22][30][36]:

$$\mathbf{X} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]^T, \mathbf{u}_k = [x_{k,1}, \ldots, x_{k,M}]^T, k = 1, \ldots, K \tag{2}$$

Vector $\mathbf{u}_k^T$ describes the $k$th user's profile as it is a set of all ratings assigned (where: T denotes transpose of the matrix **X**). Such decomposition of the matrix **X** constitutes a foundation for the users-based collaborative filtering.

It is also possible to present the matrix **X** as column vectors [6][10][22][30][36]:

$$\mathbf{X} = [\mathbf{i}_1, \ldots, \mathbf{i}_M], \mathbf{i}_m = [x_{1,m}, \ldots, x_{K,m}], m = 1, \ldots, M \tag{3}$$

where $\mathbf{i}_m$ is a set of ratings of the *m*th song assigned by all *K* users. In this case this forms a basis for representing song(item)-based collaborative filtering (this process is illustrated in Fig. 3).

Both types of collaborative filtering need further processing, i.e. in the case of the *user-based collaborative filtering* each raw denoted above is sorted by its similarity towards the *k*th user's profile. The set of similar users can be identified by employing a threshold or selecting a group of top-*N* similar users. More detailed mathematical description of this method may be found in the work by Wang *et al.* [36].

For calculating a similarity between the users *k* and *a* in the collaborative filtering typically *Pearson correlation* (Eq. 4) or *cosine similarity* (Eq. 5) measures are used, which belong to memory-based algorithms:

$$s_{\mathbf{u}}(\mathbf{u}_k, \mathbf{u}_a) = \frac{\sum_{m\in M}(x_{k,m}-\bar{u}_k)\cdot(x_{a,m}-\bar{u}_a)}{\sqrt{\sum_{m\in M}(x_{k,m}-\bar{u}_k)^2}\cdot\sqrt{\sum_{m\in M}(x_{a,m}-\bar{u}_a)^2}} \tag{4}$$

where:

– $x_{k,m}$, $x_{a,m}$ – is *m*th rate of a song assigned by the *k* and *a* users,

– $\bar{u}_k$, $\bar{u}_a$ – mean values of ratings assigned by the *k* and *a* users.

$$s_{\mathbf{u}}(\mathbf{u}_k, \mathbf{u}_a) = \cos(\mathbf{u}_k, \mathbf{u}_a) = \frac{\mathbf{u}_k \cdot \mathbf{u}_a}{\|\mathbf{u}_k\| * \|\mathbf{u}_a\|} \tag{5}$$

where:

– $\mathbf{u}_k . \mathbf{u}_a$ – scalar product of $\mathbf{u}_k$ and $\mathbf{u}_a$,

– $\|\mathbf{u}_k\|$, $\|\mathbf{u}_a\|$ – length of vectors $\mathbf{u}_k$ and $\mathbf{u}_a$ .

Reassuming, the cosine similarity is represented by a scalar product and magnitude, in the information retrieval the resulting similarity ranges are within 0 (indicating decorrelation),1 (exactly the same). These are only examples of measures and metrics that are used, *adjusted cosine similarity* is another metric employed in the ranking area. To memory-based algorithms K-Nearest Neighbor also belongs.

Other techniques such as smoothing the estimate from the collection statistics, using the linear smoothing method are employed towards derivation of the ranking formulas [6][10][22][30][36]. Apart from memory-based and model-based algorithms among CF algorithms one may discern ranking-based and probabilistic model-based [6][10][22][30][36].

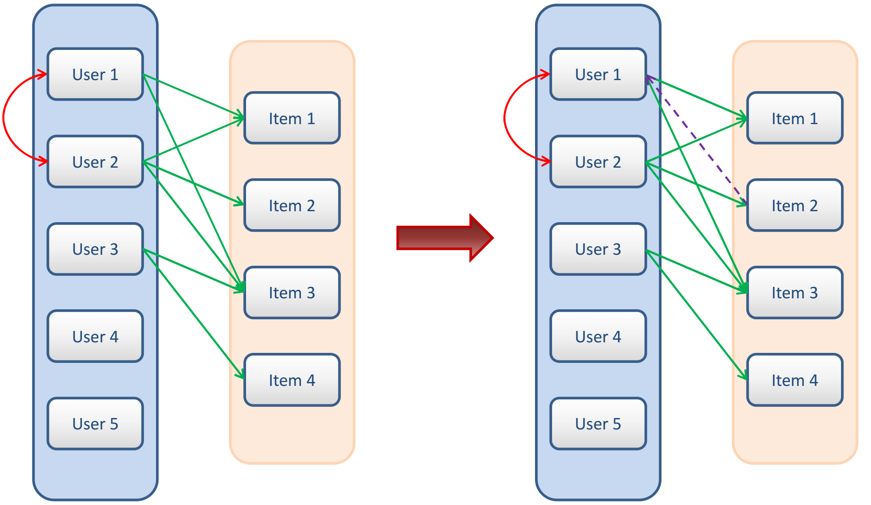


**Fig. 2.** Illustration of the user-based preference prediction for various (music) items (green lines indicate what the user listens to (and how many times), if two users (interconnected by a red arc) listen to the given song, and one of the pair listens to another one, then the implication is that the second one of the pair may want to listen to this one as well (violet dashed line)

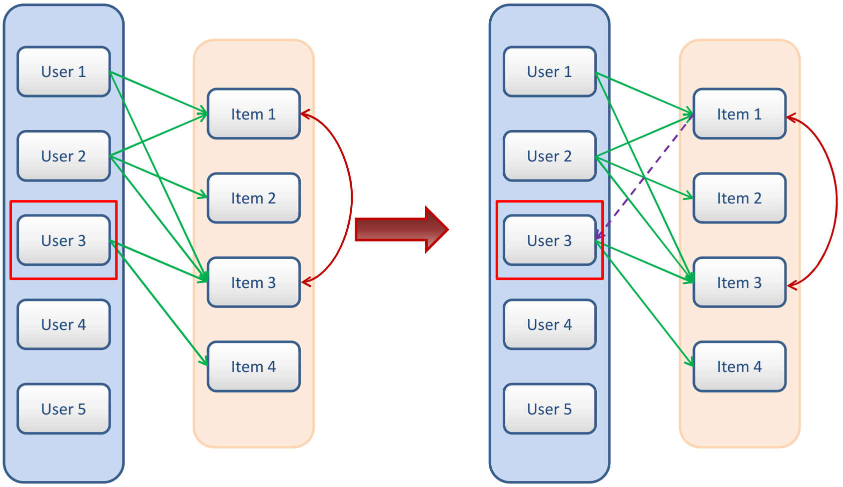


**Fig. 3.** Illustration of the item-based preference prediction (explanation as above, but concerning music items)

Due to the sparsity of the data, considering the co-occurrence statistics is unreliable. Thus in some recommender systems *Similar User Ratings*, *Similar Item Ratings* or *Similar User-Item Ratings* are used for diminishing the matrix **X**. For example in the *Similar User Ratings* approach for prediction only those songs are taken into account that were ranked highly in the ranked list, reducing the retrieval performance of the top-N returned items. In general, it is assumed that in systems with a sufficiently high user to item ratio, adding the user or changing ratings is unlikely to significantly change the similarity between two items, particularly when the items have many ratings. Therefore, pre-computing similarities between items in an item–item similarity matrix may be reasonable.

As brought by researchers working in the recommendation systems field, collaborative filtering is not risk-free. If there are millions of songs in a music service, then even very active users are not able to listen even to 1% of the music sources. This may result in unreliable recommendation. Thus, the fundamental question in modelling collaborative filtering is how to relate users and items through this usually very sparse user-item matrix.

When reviewing literature sources concerning collaborative filtering important issues related to sparsity, scalability, privacy of data, reliability, etc. are pointed out One of the very interesting ones lying at the roots of CF concerns how much confidence may be placed in the users' preferences, and whether it should be 'measured' with their consent or not. That's why the ultimate goal may be collaborative filtering without a community.

## 5 Conclusions

In this paper challenges in music retrieval and music recommendation systems were outlined. Also, reasons behind the answer to the question why and when statistics versus soft computing methods should be employed was given. Based on the review presented it may be concluded that issues of retrieval and recommendation are interconnected and these two approaches when joint together may make both processes more reliable. Also, new strategies such as for example separating music tracks at the pre-processing phase [3][8][28][29] and extending vector of parameters by descriptors related to a given musical instrument components that are characteristic for the specific musical genre to music genre classification should be more thoroughly pursued [3][28][29].

**Acknowledgements.** The study has been supported by the project no. PBS1/B3/16/2012 entitled "Multimodal system supporting acoustic communication with computers" financed by the Polish National Centre for R&D.

## References

1. Aucouturier, J.-J., Pachet, F.: Representing musical genre: A state of art. J. New Music Research **32**(1), 83–93 (2003)
2. Bergstra, J., Casagrande, N., Erhan, D., Eck, D., Kegl, B.: Aggregate features and AdaBoost for music classification. Machine Learning **65**(2–3), 473–484 (2006)
3. Eweret, S., Prado, B., Muller, M., Plumbley, M.: Score-Informed Source Separation for Musical Audio Recordings. Signal Processing Magazine, 116–124 (2014)
4. Ekstrand, M.D., Riedl, J.T., Konstan, J.A.: Collaborative Filtering Recommender Systems. Foundations and Trends in Human-Computer Interaction **4**(2), 81–173 (2011). doi:10.1561/1100000009

5. Ghias, A., Logan, J., Chamberlin, D., Smith, B.C.: Query by humming - musical information retrieval in an audio database. In: ACM Multimedia 1995, San Francisco (1995)
6. Guy, I., Zwerdling, N., Ronen, I., Carmel, D., Uziel, E.: Social media recommendation based on people and tags, pp. 194–201. ACM (2010)
7. Hoffmann, P., Kostek, B.: Music genre recognition in the rough set-based environment. In: Kryszkiewicz, M., Bandyopadhyay, S., Rybinski, H., Pal, S.K. (eds.) PReMI 2015. LNCS, vol. 9124, pp. 377–386. Springer, Heidelberg (2015)
8. Holzapfel, A., Stylianou, Y.: Musical genre classification using nonnegative matrix factorization-based features. IEEE Transactions on ASLP **16**(2), 424–434 (2008)
9. Hyoung-Gook, K., Moreau, N., Sikora, T.: MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval. Wiley & Sons (2005)
10. Konstan, J.L., Terveen, L.G., Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems Herlocker. ACM Transactions on Information Systems **22**(1), January 2004
11. Kostek, B.: Music information retrieval in music repositories. In: Skowron, A., Suraj, Z. (eds.) Rough Sets and Intelligent Systems - Professor Zdzisław Pawlak in Memoriam. ISRL, vol. 42, pp. 463–489. Springer, Heidelberg (2013)
12. Hoffmann, P., Kostek, B.: Music data processing and mining in large databases for active media. In: Ślęzak, D., Schaefer, G., Vuong, S.T., Kim, Y.-S. (eds.) AMT 2014. LNCS, vol. 8610, pp. 85–95. Springer, Heidelberg (2014)
13. Kostek, B.: Soft Computing in Acoustics, Applications of Neural Networks, Fuzzy Logic and Rough Sets to Musical Acoustics. Studies in Fuzziness and Soft Computing. Physica Verlag, Heildelberg (1999)
14. Kostek, B., Czyzewski, A.: Representing Musical Instrument Sounds for their Automatic Classification. J. Audio Eng. Soc. **49**, 768–785 (2001)
15. Kostek, B.: Perception-Based Data Processing in Acoustics. Applications to Music Information Retrieval and Psychophysiology of Hearing. Series on Cognitive Technologies. Springer Verlag, Heidelberg (2005)
16. Kostek, B., Kupryjanow, A., Zwan, P., Jiang, W., Raś, Z.W., Wojnarski, M., Swietlicka, J.: Report of the ISMIS 2011 contest: music information retrieval. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 715–724. Springer, Heidelberg (2011)
17. Li, T., Ogihara, M., Li, Q.: A comparative study on content-based music genre classification. In: Proceedings 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada, pp. 282–289 (2003)
18. Lidy T., Rauber A., Pertusa A., Inesta J.: Combining audio and symbolic descriptors for music classification from audio, Music Information Retrieval Information Exchange (MIREX) (2007)
19. Lindsay A., Herre J.: MPEG-7 and MPEG-7 Audio - An Overview **49**(7/8), pp. 589–594 (2001)
20. Mandel, M., Ellis, D.: LABROSA's audio music similarity and classification submissions, Music Information Retrieval Information Exchange (MIREX) (2007)
21. McNab, R.J., Smith, L.A., Witten, I.H., Henderson, C.L., Cunningham, S.J.: Toward the digital music library: tune retrieval from acoustic input. In: Proc. ACM Digital Libraries, pp. 11–18 (1996)
22. Mu, X., Chen, Y., Li, T.: User-based collaborative filtering based on improved similarity algorithm. In: Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology, Chengdu, China, vol. 8 pp. 76–80 (2010)

23. Ness, S., Theocharis, A., Tzanetakis, G., Martins, L.G.: Improving automatic music tag annotation using stacked generalization of probabilistic SVM outputs. In: 17 ACM Intern. Conf. on Multimedia, New York, NY (2009)
24. Pampalk, E., Flexer, A., Widmer, G.: Improvements of audio-based music similarity and genre classification. In: Proc. ISMIR, London, UK (2005)
25. Pachet, F., Cazaly, D.: A classification of musical genre. In: Proc. RIAO Content-Based Multimedia Information Access Conf., 2000 (2003)
26. Pawlak, Z.: Rough Sets. International J. Computer and Information Sciences **11**, 341–356 (1982)
27. Prechelt, L., Typke, R.: An interface for melody input. ACM Trans. on Computer Human Interaction **8** (2001)
28. Rosner, A., Schuller, B., Kostek, B.: Classification of Music Genres Based on Music Separation into Harmonic and Drum Components. Archives of Acoustics, 629–638 (2014)
29. Rosner, A., Weninger, F., Schuller, B., Michalak, M., Kostek, B.: Influence of low-level features extracted from rhythmic and harmonic sections on music genre classification. In: Gruca, A., Czachórski, T., Kozielski, S. (eds.) Man-Machine Interactions 3. AISC, vol. 242, pp. 467–473. Springer, Heidelberg (2014)
30. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proc. 10th International Conference on World Wide Web, New York, NY, USA, 285–295 (2001)
31. Schedl, M., Gomez, E., Urbano, J.: Music Information Retrieval: Recent Developments and Applications, vol. 8, no. 2–3, pp. 127–261. Now Publishers Inc., Boston (2014)
32. Schierz, A., Budka, M.: High–performance music information retrieval system for song genre classification. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 725–733. Springer, Heidelberg (2011)
33. Slaney, M., Casey, M.: Locality-Sensitive Hashing for Finding Nearest Neighbors. IEEE Signal Processing Magazine, March 2008. doi:10.1109/MSP.2007.914237
34. The International Society for Music Information Retrieval (ISMIR website). http://www.ismir.net/
35. Tzanetakis, G., Cook, P.: Musical genre classification of audio signal. IEEE Transactions on Speech and Audio Processing **10**(3), 293–302 (2002)
36. Wang, J., de Vries, A.P., Reinders, M.J.T.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proc. 29th Annual Intern. ACM SIGIR Conf. on Research and Development in Information Retrieval (2006)
37. Weinstein, E.: Query By Humming: A Survey. http://cs.nyu.edu/~eugenew/publications/humming-summary.pdf
38. http://www.amazon.com/
39. http://www.emusic.com/
40. http://www.last.fm/
41. http://musicovery.com/
42. http://www.pandora.com