Comparing Audio Features and Playlist Statistics for Music Classification

Igor Vatolkin, Geoffray Bonnin, Dietmar Jannach

Comparing Audio Features and Playlist Statistics for Music Classification

Igor Vatolkin[1], Geoffray Bonnin[2], and Dietmar Jannach[1]

[1] TU Dortmund, Department of Computer Science, Germany
{igor.vatolkin;dietmar.jannach}@tu-dortmund.de
[2] LORIA, Nancy, France, geoffray.bonnin@loria.fr

Abstract. In recent years, a number of approaches have been developed for the automatic recognition of music genres, but also more specific categories (styles, moods, personal preferences, etc.). Among the different sources for building classification models, features extracted from the audio signal play an important role in the literature. Although such features can be extracted from any digitized music piece independently of the availability of other information sources, their extraction can require considerable computational costs and the audio alone does not always contain enough information for the identification of the distinctive properties of a musical category.

In this work we consider playlists that are created and shared by music listeners as another interesting source for feature extraction and music categorisation. The main idea is that the tracks of a playlist are often from the same artist or belong to the same category, e.g., they have the same genre or style, which allows us to exploit their co-occurrences for classification tasks. In the paper, we evaluate strategies for better genre and style classification based on the analysis of larger collections of user-provided playlists and compare them to a recent classification technique from the literature. Our first results indicate that an already comparably simple playlist-based classifiers can in some cases outperform an advanced audio-based classification technique.

1 Introduction

Many studies in the research field of music information retrieval (MIR) are aimed at the automated classification or categorisation of digital musical tracks. Having the available tracks automatically categorised allows us to build better applications which, e.g., recommend music that matches the user's favorite style, help users organise their music collection based on genres, or are even capable to automatically extract semantic properties of individual musical pieces.

One of the most prominent classification scenarios is the recognition of genres and many efforts were spent on the improvement of such systems:

Sturm (2012), for example, lists several hundred references. Other categorisation goals mentioned in the literature include the identification of emotions (Yang and Chen (2012)), the recommendation of new music (Celma (2010)), or the prediction of listener tags (Bertin-Mahieux et al. (2008)); a number of further applications are described in Weihs et al. (2008).

The Music Classification Workflow

A typical algorithm chain for music categorisation comprises the following steps: (1) feature extraction, (2) feature processing, and (3) building classification models based on training examples.

Feature Extraction: As a first step, a set of typically numerical characteristics, or features, has to be chosen to represent the music data. The typical sources for the extraction of features for music data analysis are audio content, music score, music context, and user context (Serra et al. (2013)).

Feature Processing: In the second step, the extracted features are further processed. These processing steps can serve different technically required purposes like data normalisation or the imputation of missing values. In addition, feature processing steps like feature selection or transforms to lower-dimensional spaces can aim at the improvement of the classification quality or at the reduction of computation costs.

Model Building: Finally, the resulting features can be used to build classification models on some training data (labels indicating the classes of some observations). Alternatively, unsupervised learning techniques can be applied to cluster the data based on the estimated distances between data instances in the feature space.

Using Playlists for Categorisation

Building classification models from audio features is probably the most common approach in the MIR literature. When using audio signals, the extractable characteristics often describe properties of time, spectrum, cepstrum, autocorrelation, phase, etc. Music classification with audio features was applied for example in Tzanetakis and Cook (2002) or Mierswa and Morik (2005); for an overview of commonly used features see, e.g., Theimer et al. (2008), or the regularly updated manual of the MIR Toolbox (Lartillot and Toivainen (2007)).

Such approaches have the advantage that the features needed for the categorisation can be extracted from a digital music piece independently of the availability of any additional (meta-)information about it. However, relying only on audio features can have some disadvantages. First, the extraction of features from the musical signal can be computationally costly (Blume et al. (2008)). Even if these computations have to be only done once and the task can be parallelised, the sheer size of today's music collections leaves this task still challenging. Furthermore, it is often still hard to robustly extract meaningful and "interpretable" properties of the musical tracks as sometimes music with similar audio characteristics is perceived as being different by the listeners, e.g., because of their cultural background. Alternative data sources for feature extraction mentioned in the literature include for example the musical score. Such data may however be hard to obtain for all considered tracks, in particular in the area of popular music.

The recent developments in the area of online music services and music related platforms, however, opened new opportunities for researchers, as vast amounts, e.g., of user generated content annotations or listener preference information became available to be used in classification or music recommendation tasks (Hariri et al. (2012)). The work presented in this paper continues these lines of research of using user-provided (Social Web) content. Specifically, we propose to use playlists that were created and shared by users on music platforms as a data source for the classification task and present a method that relies on artist co-occurrences in the playlists to derive labelled training data. These data vectors can then be used by various machine learning techniques to build models for music classification. To the best of our knowledge, the usage of user-created playlists as input for music classification has not been explored in the literature so far. To assess the classification quality, we compare our results with those that were obtained with a recent and optimized approach that relies on the audio signal for categorisation (Vatolkin (2013)).

The paper is organised as follows. In Section 2, we describe the rationale and the technical details of our novel approach to use user-provided playlists as a source for music classification. Section 3 presents the design of our comparative evaluation and discusses the results that were observed for different musical genres and styles. In the final section, we provide an outlook on opportunities for future research in particular with respect to the combination of different data sources as was done, for example, in Lidy et al. (2007) or McKay (2010).

2 Using Playlist Statistics for Feature Extraction

Our approach is based on the assumption that homogeneity is a major quality criterion for people creating playlists as discussed in Fields (2011) and that the tracks of a playlist are correspondingly somehow similar to each other. With respect to the classification problem, we therefore assume that the presence of a given music piece in a given playlist implies a higher probability that the other songs in this list belong to the same or a similar category.

However, instead of relying on individual and possibly rare track co-occurrences, we propose to rather look at artist (composer, interpret) co-occurrences in the playlists. Given the artist of an unknown track, our goal is thus to use a machine learning model that is trained based on the information about frequently co-occurring artists for the categorisation of the track.

In the following, we describe a proposal of how to process a collection of user-provided playlists in a way that arbitrary classification algorithms like Support Vector Machines or Decision Trees can be applied. To achieve this goal, we have to derive *feature* vectors from the playlist data, which together with labelled training data points can be fed into supervised machine learning algorithms.

Figure 1 provides an overview of the steps required in our approach (top of the figure) and gives an example for the category "classic" (bottom of the figure). Our method has five steps: (1) Resolving spelling problems, (2) Identifying relevant artist co-occurrences, (3) Removing duplicates, (4) Normalisation and (5) Training of classification models.

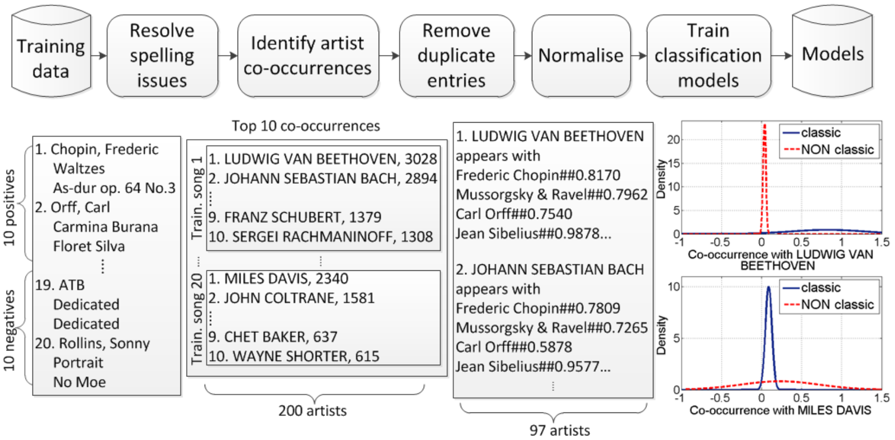

Fig. 1. Overview of algorithm steps for the extraction of playlist statistics.

Resolving Spelling Issues

A prerequisite to the computation of the co-occurrences of the tracks in the playlists is to correctly identify the tracks. As user-provided playlists often contain spelling mistakes, we applied a simple adaptation of the Smith-Waterman algorithm (Smith and Waterman (1981)) on the artist and track spellings. This algorithm was originally designed for DNA sequence alignment and computes a distance between two sequences. Applying this algorithm, we could for instance match the track name "Fragile" of Sting to the following spellings: "How Fragile", "Sting Fragile", "How Fragile We Are", etc.

Identifying Relevant Artist Co-Occurrences

The next step is to count the artist co-occurrences in the playlists in order to determine a set of "informative" artists which co-occur with other artists frequently. To do so, we iterate over each artist a of a given training set which contains tracks belonging to a music category (*positive* examples) and not belonging to it (*negative* ones)[3] and count how often (tracks of) other artists co-occur with a in the playlists. For each training track, these numbers are then sorted in decreasing order. As shown in the example, pieces created by Ludwig van Beethoven appear most often together with pieces by Frederic Chopin (3,028 times using Last.fm statistics), followed by Johann Sebastian Bach (2,894 times), and so on. Given a negative training example track for the category "classic", pieces of the artist ATB (a DJ) appear most frequently together with tracks of Miles Davis (2,340 times). Since not all co-occurrences are relevant and might introduce noise in our models, we store only the ten most frequent co-occurrences for each artist in the training dataset[4].

Removing Duplicate Entries

After the previous step and as shown in Fig. 1 we end up with a set of informative artists, which co-occurred with the artists of the 20 tracks in the training dataset that was used in Vatolkin (2013). As the same artists may appear in the top co-occurring artists lists for several training tracks (in particular for positive examples which are expected to be more similar to each other), duplicate entries in the list are removed. For the concrete example of the recognition of the genre "classic", the number of artists and their co-occurring artists – which we will later on use as *features* in the classification models – is reduced from 200 to 97 as shown in Fig. 1. We would for example see that music pieces composed by Beethoven appear frequently not only together with Chopin, but also with decreasing frequency together with pieces by Mussorgsky, Ravel, Orff, Sibelius, etc.

Normalisation

We measure the relevance of each co-occurring artists using two standard approaches based on association rules (Han and Kamber (2006)). The first approach is to use the *support* for normalisation:

$$support(\{a, b\}) = \frac{count(\{a, b\})}{N} \tag{1}$$

where $count(\{a, b\})$ is the number of playlists that contain both artists a and b and N is the overall number of playlists. Since the support values are highly dependent on the general popularity of the musical pieces, we also use the *confidence* values as an alternative:

$$confidence(a \rightarrow b) = \frac{support(\{a, b\})}{support(\{a\})} \tag{2}$$

[3] More details of the training data will be given in Section 3.

[4] In a preliminary study, increasing this number to 20 did not lead to measurable improvements. Obviously, the optimal number depends on the category; this investigation is however beyond the scope of this first study.

Training of Classification Models

Based on the normalised co-occurrences with the artists from the training set (the co-occurrences values serve as features) and the given category assignments, classification models can be finally built using different machine learning approaches. For instance, Naive Bayes predicts classes based on feature distributions for positive and negative instances. An example of the density of the feature distribution is provided in the right hand side of Fig. 1. Tracks that do not belong to the “classic” genre appear very seldom together with Beethoven, which is indicated by the high peak of the density function for values close to zero. On the other hand, there are only a few classic pieces which appear together with tracks of Miles Davis.

At the end, after the models have been trained, they can be applied for the classification of unlabelled tracks for which the artist is known using the chosen machine learning technique.

3 Experiments

3.1 Experimental setup

To be able to compare our playlist-based approach with a typical audio signal based one, we used the experimental setup from Vatolkin (2013), where the goal was to categorise music tracks into 6 genres (Classic, Jazz, Pop, etc.) and 8 styles (e.g., ClubDance, HeavyMetal, Urban) using binary classifiers.

Dataset

For each of the 14 categories, the dataset comprises 10 positive examples and 10 negative ones. In addition, Vatolkin (2013) used an optimisation set of 120 tracks to apply an evolutionary feature selection technique in order to determine the most relevant audio features for learning. The models were then evaluated on a test set which also comprised 120 tracks and which had the same genre distribution as the optimisation set.

Audio Features

We use four sets of audio features after Vatolkin (2013). The first group describes 636 low-level audio signal characteristics. The second group consists of 566 high-level “semantic” descriptors, which are better interpretable like, e.g., the recognised instruments, moods, harmonic properties, etc. The third group contains 13 Mel Frequency Cepstral Coefficients (MFCCs) which were developed for speech recognition but are commonly used in music classification (Meng et al. (2007)). The fourth group contains the optimised feature sets after the application of an evolutionary feature selection strategy.

Playlist Features

For the four groups of playlist statistics, we used two data sets retrieved from public sources[5] and the two normalisation methods described in Equations 1 and 2.

Classification and Evaluation

As classification techniques, we used Decision Tree C4.5, Random Forest, Naive Bayes, and Support Vector Machines. In the following section, we report the results of the method that worked best for the specific classification task. Because the distribution across genres and especially across styles is not balanced, classification models are evaluated with the balanced relative classification error:

$$e_{BRE} = \frac{1}{2}\left(\frac{FN}{TP+FN} + \frac{FP}{TN+FP}\right), \tag{3}$$

where TP denotes true positives, TN true negatives, FP false positives, and FN false negatives.

3.2 Results

General Results

The classification errors obtained in the experiments for the 8 feature sets and the 14 categorisation tasks using the classification method leading to the best results[6] are shown in Fig. 2. When looking on the audio-based approaches (symbols with white background), the feature optimisation method of Vatolkin (2013) not surprisingly worked best except for the category "Jazz" (for this category, the validation set contained more European Jazz and the optimisation set more American Jazz).

To some surprise, however, the comparably simple classification method based on playlist statistics and artist co-occurrences performs equally well and in many cases even better than the method based on optimised audio feature sets. The best variant of the playlist-based methods outperforms the best audio-based approach for 10 of 14 categories. This indicates that the computationally highly efficient and rather simple aggregation of playlist statistics can be indeed a good alternative for music classification. For some categories, however, audio features performed better. The MFCC-based feature set was for example particularly successful for the classification of Rap music. These results therefore suggest the use of hybrid strategies that combine the different approaches.

[5] The samples included about 1 million playlists from Last.fm and about 600,000 playlists from 8tracks, see also Bonnin and Jannach (2013).

[6] The best performing method depends on the category. Moreover, the removal of a weaker classifier from ensemble of above mentioned methods led to a statistically significant reduction of performance in a previous study (Vatolkin et al. (2014)).

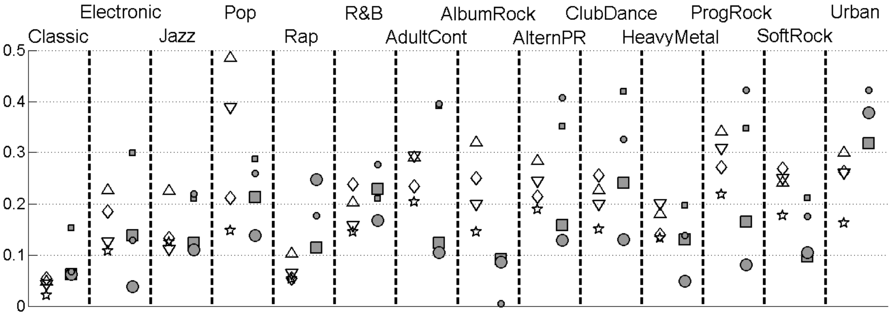

Fig. 2. Balanced relative classification errors for 14 music categories (labels above the figure) and 8 feature sets. Audio features, signs with white background: Downward-pointing triangles: low-level features; upward-pointing triangles: high-level features; diamonds: MFCCs, asterisks: optimised feature sets. Playlist features, signs with shaded background: rectangles: 8tracks; circles: Last.fm; larger signs: normalisation with confidence; smaller signs: normalisation with support.

Further Observations

The normalisation based on confidence generally performs better than when using the support statistic for the 8tracks data in 13 of 14 cases, and for Last.fm in 12 of 14 cases. Furthermore, the mean performance on the Last.fm dataset is generally higher than for 8tracks (in 10 of 14 cases). This can be simply explained by the larger amount of data that is available in the used playlist collection of Last.fm.

Another outcome of the study is that the obtained classification quality varies with the different classification methods. Playlist-based approaches seem to often perform slightly better if the models are trained with a Naive Bayes approach or Support Vector Machines. A systematic tuning of the hyperparameters of the classification methods has not yet been done but may help to further increase a classification performance. Another improvement could potentially be achieved if a feature optimisation strategy would also be applied to the playlist-based approach.

4 Conclusions and Outlook

In this work, we investigated how well two methods for the aggregation of playlist statistics are suited to build feature sets for genre and style classification. We compared the classification quality of using playlist statistics with the quality that can be achieved when using classification models based on optimised audio feature sets. Our results showed that playlist-based models were favourable over audio-based features sets for classification for more than half of the genres.

The choice of which features to use in real-world classification-based applications in our view strongly depends on the main guiding constraints in the goal of the particular application setting. Consider the following example scenarios.

1. If the application's goal is to derive interpretable harmonic and melodic properties, e.g., of user-defined personal categories, a music scientist would probably prefer automatic classification based on high-level audio features as playlist-based models do not operate on the basis of such features.
2. In case that the processing efficiency for the classification task is the main requirement, e.g., because huge music collections have to be analysed, one might prefer playlist-based models as they help to avoid the computationally costly extraction of features from the audio signal.
3. If the quality of the classification is the most important application requirement, a combination of audio features and features derived from playlist statistics might be the best choice.
4. Finally, for researchers, using playlist information in our view represents a comparably cheap way of developing classification approaches with competitive performance, because the musical tracks themselves do not have to be purchased or licensed for the analysis.

As part of our future work, we plan to examine the performance of combined feature sets where we also aim to apply feature selection techniques that simultaneously consider the feature sets of both sources. In addition, the validation of such an approach is planned using other public data sets.

Another promising direction for further research in our view is the development of further variants of our playlist-based classification methods and the evaluation of various parametrisations of the techniques. Specifically, this could involve the integration of statistics from other web sources, the systematic variation of individual parameters like the number of the stored top co-occurrences, the consideration of track and album co-occurrences, or the fine-tuning of the underlying classification methods.

References

BERTIN-MEHIEUX, T., ECK, D., MAILLET, F., and LAMERE, P. (2008): Autotagger: A Model for Predicting Social Tags from Acoustic Features on Large Music Databases. *Journal of New Music Research*, 37(2):115-135.

BLUME, H., HALLER, M., BOTTECK, M., and THEIMER, W. (2008): Perceptual Feature based Music Classication - a DSP Perspective for a New Type of Application. In *Proc. IC-SAMOS*, 92-99.

BONNIN, G. and JANNACH, D. (2014): Automated Generation of Music Playlists: Survey and Experiments. *ACM Computing Surveys*, 47(2).

CELMA, Ò. (2010): *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play In the Digital Music Space.* Springer.

FIELDS, B. (2011): *Contextualize Your listening: the Playlist as Recommendation Engine.* PhD thesis, University of London.

HAN, J., KAMBER, M. (2006): *Data Mining: Concepts and Techniques.* The Morgan Kaufmann Series in Data Management Systems.

HARIRI, N., MOBASHER, B., and BURKE, R. (2012): Context-Aware Music Recommendation Based on Latent Topic Sequential Patterns, *Proc. ACM RecSys 2013*, 131-138.

LARTILLOT, O. TOIVAINEN, P. (2007): MIR in Matlab (II): A Toolbox for Musical Feature Extraction from Audio. In: *Proc. Int'l Conf. on Music Information Retrieval (ISMIR)*, 127-130.

LIDY, T., RAUBER, A., PERTUSA, A., and IÑESTA. (2007): Improving Genre Classification by Combination of Audio and Symbolic Descriptors Using a Transcription System. In: *Proc. Int'l Conf. on Music Information Retrieval (ISMIR)*, 61-66.

MCKAY, C. (2010): *Automatic Music Classification with jMIR.* PhD thesis, McGill University.

MENG, A., AHRENDT, P., LARSEN, J., HANSEN, L. K. (2007): Temporal Feature Integration for Music Genre Classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1654-1664.

MIERSWA, I. and MORIK, K. (2005): Automatic Feature Extraction for Classifying Audio Data. *Machine Learning Journal*, 58(2-3):127-149.

SERRA, X., MAGAS, M., BENETOS, E., CHUDY, M., DIXON, S., FLEXER, A., GÓMEZ, E., GOUYON, F., HERRERA, P., JERDA, S., PAYTUVI, O., PEETERS, G., SCHLÜTER, J., VINET, H., and WIDMER, G. (2013): *Roadmap for Music Information Research.* Technical Report, The MIReS Consortium.

SMITH, T., WATERMAN, M. (1981): *Identification of Common Molecular Subsequences.* Journal of Molecular Biology, 147:195-197.

STURM, B. (2012): A Survey of Evaluation in Music Genre Recognition. In: *Proc. Adaptive Multimedia Retrieval (AMR).*

THEIMER, W., VATOLKIN, I., and ERONEN, A. (2008): *Definitions of Audio Features for Music Content Description.* Technical Report TR08-2-001, TU Dortmund.

TZANETAKIS, G. and COOK, P. (2002): Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293-302.

VATOLKIN, I. (2013): *Improving Supervised Music Classification by Means of Multi-Objective Evolutionary Feature Selection.* PhD thesis, Department of Computer Science, TU Dortmund.

VATOLKIN, I., BISCHL, B., RUDOLPH, G., and WEIHS, C. (2014): Statistical Comparison of Classifiers for Multi-objective Feature Selection in Instrument Recognition, *Data Analysis, Machine Learning and Knowledge Discovery.* Springer, 171-178.

YANG, Y.-H. and CHEN, H. H. (2011): *Music Emotion Recognition.* CRC Press, Boca Raton.