NOKIA

Research Center
NRC-TR-2007-012

Introduction to Methods for Music Classification Based on Audio Data

Igor Vatolkin* and Wolfgang Theimer

Nokia Research Center Bochum, Germany
University of Dortmund, Germany*
http://research.nokia.com
August 24, 2007

Abstract:

The subject of music information retrieval (MIR) is to analyze and categorize music pieces. Over the last years many approaches have been designed to automatically extract music data from the digitized audio signal. This article presents a survey of the state-of-the-art algorithms on the basis of a broad literature study and a tool analysis. It should help to navigate through different MIR techniques and tools. An overview of different music features to characterize timbre, harmony, melody and rhythmic information is given. The various time scales of feature extraction to form meta-features from basic features are discussed. The task-specific pruning of features is presented to reduce the computational complexity. The article continues with a discussion of different classification techniques and how the results are evaluated. Finally the properties of four state-of-the-art MIR tools are outlined.

Index Terms:

music information retrieval
music feature extraction
feature pruning
music classification

Introduction to Methods for Music Classification Based on Audio Data

Igor Vatolkin and Wolfgang Theimer

August 24, 2007

Abstract

The subject of music information retrieval (MIR) is to analyze and categorize music pieces. Over the last years many approaches have been designed to automatically extract music data from the digitized audio signal. This article presents a survey of the state-of-the-art algorithms on the basis of a broad literature study and a tool analysis. It should help to navigate through different MIR techniques and tools. An overview of different music features to characterize timbre, harmony, melody and rhythmic information is given. The various time scales of feature extraction to form meta-features from basic features are discussed. The task-specific pruning of features is presented to reduce the computational complexity. The article continues with a discussion of different classification techniques and how the results are evaluated. Finally the properties of four state-of-the-art MIR tools are outlined.

1 Introduction and Motivation

In recent years, the presence of digital music has become ubiquitous. The expansion of the internet, music online shops and the enhanced storage capacities of stationary and mobile devices such as music players, smartphones, PDAs caused a remarkable growth of private music collections all over the world. Since these collections often exceed thousands of music pieces, the importance of intuitive content navigation is growing constantly. The common approach to manage large data sets is to classify them, often hierarchically, and to use keywords for the description of data. However, the existing definitions of music categories, typically called genres, are ambiguous and very hard to standardize. In [1] it is shown that taxonomies from three popular music classification web sites http://www.allmusic.com (531 genres), http://www.amazon.com (719 genres) and http://www.mp3.com (430 genres) share only 70 equal genre definitions. The definition of one genre or music style depends on many facts such as instrumentation, place of the genre formation, cultural and historic information. Besides that, a possibly unique genre taxonomy list can be meaningless for individual persons. For a lover of classical music whose music collection consists of several thousand classical and several dozens of rock and jazz pieces, 'Popular Music' can describe the small part of his collection very well. A jazz fan could distinguish song categories such as Acid Jazz, Latin Jazz, Swing Jazz, Bebop and so on. Furthermore, many personal genre categories which depend on the preferences of their owner can be used, e.g. 'my favorite songs', 'sad music', 'summer holidays'. So, if the goal of a music management tool is to assist the user in navigation through a large music collection, user-driven classification is essential. Another aspect of music classification is the fuzziness of suggestions such as 'music piece A belongs to music genre B'. Due to the unprecise boundaries between genres it is meaningful to allow one music piece to belong to several music categories with different fuzzy membership values.

Beside the mapping of music pieces to pre-defined categories some other music tasks can be performed by computer algorithms developed during the last decades. These problems are e.g. the recognition of particular instruments, the recognition of hummed melodies, the extraction of complex music descriptors like tonality or structural information, the beat and similarity analysis.

Several information sources exist to categorize music: Features can be extracted from the audio signal, features are derived from symbolic representations like MIDI or the musical score and the community information like user-generated playlists or recommendations are utilized. Since the availability of scores

for popular music is very limited for the end user, and the community information can be sometimes unprecise or subjective, this paper concentrates on the classification based on audio data. Current methods and several tools for music feature extraction and classification are described and put into perspective. Proposals and ideas for future developments are presented.

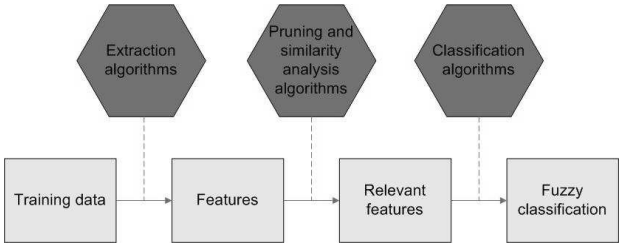

Figure 1: Processing steps

2 Processing Steps for Music Classification

A successful music categorization is based on several algorithmic steps, as depicted in Figure 1. First of all, a given feature set is to be extracted from the set of music pieces. These features describe different characteristics of music pieces, e.g. physical, temporal, harmonic or even cultural properties. The number of extracted features can be very high. In order to reduce the computational complexity, the feature values are analyzed and only important features are chosen for classification algorithms. This task is covered by pruning and similarity analysis algorithms, called feature pruners in this paper later.

Normally, the given data set is split into training set and test set: Classification algorithms first learn to classify the given music data and then a validation shows how good the classification method works.

A variety of tools for MIR and especially automatic classification of music pieces has been developed in recent years. The different tools are either available as open-source or as commercial offers. They have different focus areas: Feature extraction, classification or visualization of music. For this paper we have tested several commonly used open-source tools, which are continuously developed and represent the state-of-the-art. Dedicated math packages like e.g. Matlab are left out here due to their general purpose nature, although they are used intensively also in the music domain. The exact descriptions of tools are listed below in the appendix. For the extended list of further tools, see [2] and the web site of the author [1].

The algorithms for every processing step are described in detail in the next chapters.

[1] http://mirsystems.info/

3 Music Feature Extraction

A digital music signal is a special case of a value series (see [3] for definition) since it is characterized by a time sequence of sampled and quantized audio values. In the context of this article it is assumed that multichannel audio information is reduced to a mono signal which contains all relevant music information for classification.

3.1 Timbre features

Timbre is the perceptual feature that makes two sounds with the same pitch sound different. It describes the spectral content of music and provides the summary information of all instruments and voices at the same time. Thus, this feature can be used for the distinction of instruments.

- *Temporal features* are computed from single signal intervals or from the complete signal. Important features are the zero-crossing rate, linear prediction coefficients as well as variance and mean distance between extremal or zero values [4].

- Another type of features describes the *energy information* of the signal. Important parameters are the root mean square and absolute values of a signal frame (volume), its variation over time, the energy of harmonic components, the energy of the noisy part of the spectrum and low energy segments [5]. The odd-to-even harmonic energy ratio gives the ratio between odd harmonics energy to even harmonics energy and can be used to distinguish instruments [6]. The tristimulus consists of three values. The first one corresponds to the spectral energy ratio between the fundamental frequency and the sum of all harmonics; the second corresponds to the ratio between the sum of the second, third and fourth

harmonics and the harmonic sum; the third specifies the ratio of fifth and all further harmonics to the harmonic sum [7].

Several measures exist to judge the purity of the timbre information [6]: The noisiness is defined as the ratio between the noise (non-harmonic part of the spectrum) and the complete spectral energy. It is close to 1 if the signal is dominated by noise and around 0 if the harmonic content dominates. The inharmonicity represents the deviation of the spectral components from a pure harmonic structure. The harmonic spectral deviation describe how the amplitudes of harmonics peaks differ from the global spectral envelope. In [8] it is shown that the harmonic components of a sound alone are sufficient to perform instrument recognition, but the non-harmonic noise improves the recognition and the naturalness of the sound.

- *Spectral shape features* describe the shape of the power spectrum on a more abstract level. The centroid [5] describes the center of gravity of the magnitude spectrum, the parameters spread / bandwidth, skewness, kurtosis [6] as well as linear regression features such as spectral slope [4] characterize the extent of the spectrum. The spectral flatness is the ratio of geometric and arithmetic spectral mean values. If the spectrum is balanced, i.e. no dominant spectral peaks exist, the spectral flatness value is near zero. The spectral crest factor [4, 6] provides the ratio between maximum and average spectral values. Mel frequency cepstral coefficients (MFCCs) are calculated from the cepstral audio representation (the spectrum of the spectrum) and consider the human sound recognition using the logarithm of the amplitude spectrum [9]. Variants of the cepstral coefficients use the bark scale (BFCCs), the equivalent rectangular bandwidth scale (EFCCs) or the octave scale (OFCCs) [10]. Other relevant features are spectral flux (describing the spectral change in successive intervals) [5, 6], spectral decrease [6], roll-off frequency [5] and maximum frequency and variance in defined intervals [10].
- *Phase domain features* attempt to model the dynamics of a nonlinear system by creating vectors from a state variable. The angles between adjacent vectors and their distances / variances allow the classification of certain music genres [3].
- *Perceptual features* have been introduced to model the human acoustic perception. Existing features are transformed to mimic the human hearing: Relative loudness, sharpness and spread of sound [6]. The Bark or Sone representation on the frequency scale allow to model the human perception that is most sensitive to medium frequencies between 1 - 5 kHz and less sensitive to lower or higher frequencies [10].

3.2 Harmony and melody features

Harmony is defined as the usage of simultaneous pitch values and chords in music. Since the associated notes appear vertically in a musical score, harmony is called the vertical element of music. Melody is defined as the succession of pitched events which are perceived as a single entity. Therefore it is entitled the horizontal element of music [11].

- The *pitch* as the fundamental frequency of a sound is a key feature for melody and harmony analysis. The fundamental frequency is defined as the frequency whose integer multiples best match the spectral content of a signal [6]. Pitch distributions (either single or multiple fundamental frequencies at a time) with their amplitudes, width and frequency positions / distances of peaks are retrieved [4].
- *A chromagram* maps all pitch values to a frequency range of one octave by a modulo operation (folded pitch) and allows to sense the harmonic contributions. By construction it is invariant against transposing music by multiples of an octave. Also a mean chroma vector is used by several authors [10, 12]. In order not to loose information the pitch distribution or histograms over a number of time windows are created. The most dominant parameters from those histograms are the amplitude and the pitch value of the maximum peak of the folded or unfolded pitch distribution, the pitch interval between the two most prominent peaks and the sum of all pitch values as a measure of strength for the pitch detection [5].
- The *tonality* (key and mode) of a music piece describes the relationship between simultaneous (multi-pitch) and successive (melody) tones. Its estimation is a very complex task, the common approach uses hidden Markov models since the transitions of notes and chords can well be described by probabilities. The relationship be-

tween different tones in a given tonality was investigated by experiments [13, 14] and their results are used for algorithms which estimate key and mode on the basis of chroma vectors [15, 16]. A shortcoming of most algorithms is the limitation of classification to major/minor modes. Music pieces in other modes such as dorian or phrygian as well as music outside the Western tradition (e.g. pentatonic scale, quarter tone music) cannot be analyzed adequately.

3.3 Rhythm and time properties

Pieces of music can also be distinguished by their bar and temporal structure. Different rhythmic patterns and accents correspond to different music styles.

- The most trivial time measure is the duration of a piece of music which can act as a simple means to filter different pieces of music.
- In general music contains nested groups of pulses on different time scales, called metrical levels. Time intervals on higher metrical levels are integer multiples of the lower level periods. The lowest metrical level is termed *tatum*.
- The *beat* is the perceived periodic variation in loudness, resulting from acoustical interference of two near-unison tones [17]. A typical beat range is between 40 and 200 beats / minute. Since it is sometimes difficult to estimate the exact beat frequency, it is common practice to create beat histograms and summarizing features such as amplitudes / positions of the first two peaks, the ratio of the amplitudes of second and first peak or the overall sum of all beat values to indicate the strength of the beat altogether. The beat is typically evaluated by applying prefiltering and autocorrelation to the signal and averaging the beat values over a long time window [5]. A method for joint efficient computation of beat and tatum information based on a subband approach and transformation into the periodicity domain is described in [18].
- The *rhythm* of a piece of music describes the temporal regularity pattern governing the music. Periodicities are obtained by autocorrelation of features over time, like e.g. energy in different frequency bands or MFCCs [4]. The time scale of rhythms is longer than that for beats. In music notation it is typically described by a musical meter like e.g. 3/4 or 4/4. A successful approach to estimate a periodic rhythmic pattern and the music meter is given by [19]: The authors use a Bayesian probabilistic model to map an input signal to a cycle of the rhythmic pattern and are also able to track changes in rhythmic patterns, meter and tempo.
- The rhythm can be quantized to belong to a certain category (e.g. by a musical meter). Also the (intentional) timing deviations from a regular rhythm characterize a piece of music.

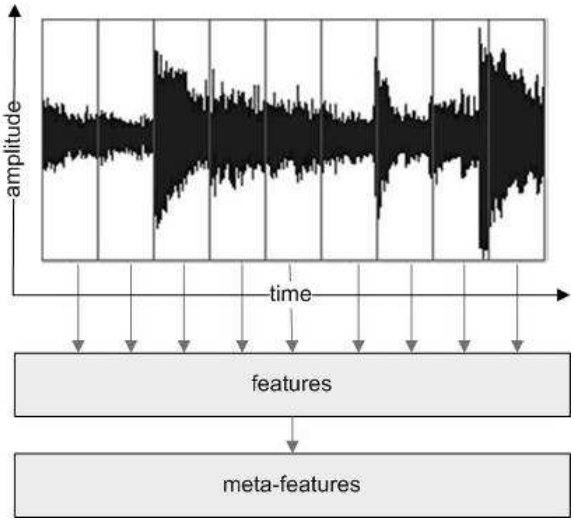

Figure 2: Hierarchical combination of features

3.4 Implementation of feature extraction and aggregation of features

The features described above are extracted from a particular segment of music data, for example a time window with 512 samples. The number of samples should be small enough to not correspond to several melody tones, but also large enough for spectral estimations. If the whole music piece is divided into time windows of a given length, the variation of every feature over time can be analyzed. This process can be characterized as the building of meta-features or high-level features (see Figure 2).

Several possibilities exist to build meta-features. Let $f_{1,\ldots,n}$ be the extracted features and t the number of time windows.

- The standard *statistical descriptors* are minimum, maximum, median and mean values and the

standard deviation. Moments describe the distribution of feature values [20]. The most significant moments are mean, variance, skewness and kurtosis. The feature values can be sorted and divided into a given number of quantiles, groups with equal number of values. Also, the value range between minimum and maximum can be divided into a given number of histogram bins, and the strength of each bin can be calculated.

- Another type of meta-features models *time dynamics*. Running means or (n-th) derivatives of feature values are examples to be calculated. Hidden Markov models (HMMs) [21] describe the probabilities of transitions between states, where each state corresponds to a defined range of values. Other descriptors of dynamics were calculated e.g. in [22].

The number of values for each meta-feature varies between 1 (mean value, moments...) and t (derivative e.g.), i.e. the number of time windows.

However it is common practice not to extract features from adjacent time windows but from sliding time windows with overlap (e.g. 50%). In that case the overall number of features increases by 50 %. The extraction of these values and the additional training of classification strategies based on all feature values requires a lot of computing time. Therefore some considerations which reduce the number of features before estimation of meta features (described later as feature pruning) are used in most audio classification tools. Often the following signal processing steps are applied:

- Reduction from stereo to mono audio signal.
- Sampling rate conversion (e.g. to a standard sampling rate of 22.05 kHz as in MusicMiner [10]).
- Determination of segments for feature extraction: Often the introduction is skipped since it is transient, a longer segment (e.g. 30 s) is selected some time after the beginning (e.g. after 30 s), feature extraction is only done for sung parts or for the complete piece of music. By using a self-similarity or distance matrix recurrent structures can be identified and only those segments are used for feature extraction [23]. Another possibility is the choice of a given number of time windows from random positions, applied for example in [24].
- Sampling can be applied for feature extraction, i.e. features are evaluated in short time windows distributed over the piece of music.
- A sliding window strategy defines the window size (e.g. of 512 samples or 20 ms length) and the overlap (e.g. 50%) with the next window.

4 Feature Pruning

In contrast to the discussion of music features or classification strategies, the methods to limit the number of used features are not so well represented in literature. The common reason is that authors usually start classification with a small given set of important features and do not need to reduce this set any more. However, different classification tasks may require different features or feature groups, so it is not always reasonable to limit the number of used features from the start. An adequate approach here is to take a large amount of possible features into account and use feature pruners for the estimation of the most important features. In other words, the task of a feature pruner is to choose a small feature set which is good enough to allow successful classification. Commonly used pruning strategies are:

- *Linear reduction of the feature space dimensions*: Here the whole set of features is needed, and the axes are re-defined. The *principal component analysis* (PCA) [25] calculates the eigenvectors of the covariance matrix of feature vectors and transforms the axes in the feature domain, so that they can be sorted by the variance of features along the axes. The most important axis corresponds to the largest feature variance along this axis. Axes with the smallest feature variances are discarded and the dimensionality of the problem is reduced. Another linear technique is *linear discriminant analysis* (LDA) [26], which transforms the axis on the basis of scatter matrices. The goal is to reduce the distances within feature vectors which belong to the same (music) category and to increase the distances between different categories.
- The *correlation-based feature selection* [27] begins with an empty set of features and adds them one by one, selecting at first the features which are least correlated with the features already added to the set.

- *Pareto density estimations* (PDEs) used for feature selection in [10] calculate the likelihood of a membership in a certain music category for given feature values. In other words, the likelihood is described as a function of feature distributions. Several quality scores for features are introduced: Separation scores e.g. measure the area under PDEs corresponding to the error which may occur when estimating the category.

- *Evolutionary algorithms* (EAs) [28] consider several aspects of natural evolution and produce new sets of solutions from the existing ones with defined operators, so that the positive characteristics of parent solutions are transfered to the next generation. For feature pruning, a solution describes the composition of a feature set. The quality of a certain solution corresponds to the success of classifiers which have been run on this feature set. [29] mentions several drawbacks of statistical techniques such as PCA and proposes genetic algorithms to be a promising method to reduce feature dimensionality.

- Nonlinear dimension reduction techniques and neural networks are not so often used in MIR for feature pruning, but are applied in other problem domains.

5 Classification and Evaluation of Results

The purpose of classification algorithms is to map music songs to categories, whereas one song can be member of several categories. Clusters in the feature domain can be built, or the similarities between given instances are calculated based on similarity metrics. Since the ranges of feature values can be very different, normalization should be done before classification (e.g. mapping of features to interval [-1;1] or [0;1]). Alternatively the results of independent classifiers for different sets of features should be combined.

5.1 Similarity metrics

Several metrics are used by classification strategies for measuring the similarity of different feature vectors. Let $\mathbf{f}_x$ and $\mathbf{f}_y$ be two feature vectors with f_x^i and f_x^i being their i-th scalar components. Most common used metrics are:

- The *Euclidian distance* is the standard distance measure:

$$d_e = \sqrt{\sum_{i=1}^{n} (f_x^i - f_y^i)^2} \tag{1}$$

- The *cosine distance* corresponds to the cosine of the angle between two vectors. If two vectors have the similar direction, the similarity is maximum (equal to 1):

$$d_{cos} = \frac{\mathbf{f}_x^T \mathbf{f}_y}{||\mathbf{f}_x|| \cdot ||\mathbf{f}_y||} \tag{2}$$

- *Hamming distance*: The standard Hamming distance calculates for two feature vectors the number of positions with different (discrete) values. Continuous feature values may lie within some defined neighborhood, e.g. interval and can be then described as similar. Here one can sum the number of non-empty neighborhoods.

- The *Mahalanobis distance* considers the correlation between features and is scale-invariant:

$$d_{mah} = (\mathbf{f}_x - \mathbf{f}_y)^T \cdot C^{-1} \cdot (\mathbf{f}_x - \mathbf{f}_y) \tag{3}$$

  with the covariance matrix

$$C = E\left[(\mathbf{f}_x - E[\mathbf{f}_x]) \cdot (\mathbf{f}_y - E[\mathbf{f}_y])^T\right] \tag{4}$$

Beyond the metrics listed here many other methods to calculate the similarity between two feature vectors exist, for example self organizing maps (SOM), a technique based on artificial neural networks which maps similar feature vectors to close positions on a two-dimensional map [30] or the Kullback-Leibler divergence which considers probability distributions and is used in [31].

5.2 Classification strategies

Many classification strategies were developed for data mining. For details and an overview of different techniques, see [32]. In the music classification context, some categories are defined in the beginning and examples of music are assigned to them. Each piece of music can be mapped to none, one or multiple of the defined categories. The purpose of the classifiers is to learn from the given data and map new music pieces to the correct music categories. These classifiers belong to *supervised learning* algorithms, which

learn from the given training set. *Unsupervised learning* algorithms start with no information about the output categories at all and try to build the categories by themselves (e.g. clustering). These strategies are not often used for music classification. We provide here a list of the commonly used algorithms:

- *k-nearest neighbors*: Each song described by feature vector can be represented as a point in the multidimensional feature space. For a new song the distances to the nearest neighbor from each category are calculated. The song is assigned to the category whose nearest neighbor is closest. Different distance metrics as discussed above can be used. It is also possible to make a fuzzy classification. The strategy is simple, but effective and can be enhanced with a special data structure, kD-trees, which recursively divide the feature space with hyperplanes.

- *Bayesian classifier*: The probability distribution of feature values X for a certain category Y is estimated, e.g. modelled as a Gaussian distribution. On the basis of probability densities for all categories, the probability that the new song belongs to a certain category can be computed. The Bayesian theorem can be used to compute the conditional probability that a category Y is observed for given features X:

$$P[Y|X] = \frac{P[X|Y] \cdot P[Y]}{P[X]} \qquad (5)$$

  The individual features are treated as if they were independent, and the probability densities are multiplied. Normally, the features are not independent and some of them correlate, so the calculation of the probability belonging to one certain category is not precise. Therefore a pruning operation to reduce the correlation between different features is beneficial to increase the classification performance.

- The *divide-and-conquer* algorithm makes a classification on the basis of *decision trees*. Decision trees consider one feature in each node. The child nodes correspond to some intervals of the feature values. For example, a simple decision tree can start in a 'number of beats per minute' node and proceed to three child nodes, the first for interval (0;60] beats/minute (classical music), the second for interval (60;100] bpm (pop music) and the last for bpm values greater than 100 (electronic music). Divide-and-conquer estimates the importance of every feature on the basis of an entropy measure, which describes how much information about the membership to different categories will be gained if this feature will be taken as a node. The most important feature is taken initially as the first node, and the algorithm recursively continues for the child nodes.

- *C4.5*: Several enhancements of divide-and-conquer lead to the algorithm named C4.5 [33]. The most significant improvement is the pruning of large decision trees. A commercial successor of C4.5 exists (C5.0) which works more efficiently, but is not available as open source and is not widely used in music information retrieval.

- *Support vector machines* (SVMs): If the points in feature space cannot be linearly separated, SVMs create new feature dimensions and transform the points so that they can be separated with maximum margin. Further modifications include the usage of nonlinear bounds. An extended tutorial to SVMs is given in [34].

If no information about music categories is available, unsupervised methods like *k-means clustering* can be used. Here k cluster centers (corresponding to the categories) are randomly picked up from the feature vectors, and the other feature vectors are mapped to the clusters on the basis of some distance measure, often using Euclidean metric. Once the input feature vectors are assigned to the respective clusters, the cluster centers are calculated again as the centroids of all feature vectors from the cluster and the whole process is repeated with the new cluster centers. Another unsupervised learning method is the *Gaussian mixture model*, which describes the feature vector as a combination of Gaussian distributions [35].

One important conclusion from the literature study is that the choice of classification strategy is often less important than the design of the feature set. As a single classifier, SVMs perform well and are often preferred for music classification. Another interesting idea is to use several classifiers and combine their results, for example running different classifiers on different feature groups [36].

5.3 Evaluation of results

Evaluation of classification results belongs to the design steps of a music classification system like the

choice of a feature set and a classification strategy. Generally evaluation is a bottleneck in data mining since the algorithms often are trained on a small set of data. Larger data sets are not always available and require much more learning time. The classifiers should perform well for some new data which was not known during the learning process. Well-described data for learning is normally divided into training and test sets, so that the algorithms can learn on the training set and are evaluated on the test set. The methods to arrange training and test set are:

- *m-fold cross validation*: As a very common used technique (applied e.g. in [3, 5, 37]) the data set is divided in m equal partitions. During a single processing iteration, all partitions except one are used for learning and the last partition is used for evaluation. The validation is repeated exactly m times for different partitions, so that every partition is used once for evaluation and $m-1$ times for learning. The whole m-fold cross validation can be repeated several times and the evaluation results are averaged.

- *Leave-one-out*: m-fold cross validation, where the number of partitions is equal to the number of data instances.

- *Separated training and test sets*: In [38] it is argued that cross validation can lead to the overfitting of algorithms to the given data set and therefore separate training and test sets should be used.

Some measures for the quality of classification can be considered:

- *Number of successes and failures*: TP_i (true positives) is the number of music files that belong to category i and are recognized correctly; FN_i (false negatives) is the number of music files which belong to category i but are identified as not belonging to the category; TN_i (true negatives) corresponds to the amount of music pieces which do not belong to category i and are classified as not belonging to the category; FP_i (false positives) is the number of music pieces which do not belong to category i but are classified as belonging to it.

- *Precision* describes the fraction of correctly identified music pieces of category i to the whole set of music pieces identified as belonging to category i:

$$p_i = \frac{TP_i}{TP_i + FP_i} \cdot 100\% \tag{6}$$

- *Recall* describes the fraction of correctly identified music pieces of category i to the whole number of music pieces of category i:

$$r_i = \frac{TP_i}{TP_i + FN_i} \cdot 100\% \tag{7}$$

- *Accuracy* corresponds to the average rate of true positives. Let C be the number of categories and N the whole number of music pieces:

$$a = \frac{1}{N} \cdot \sum_{i=1}^{C} TP_i \cdot 100\% \tag{8}$$

- F_1 *measure* is a combination of precision and recall described in [37]:

$$F_1 = \frac{2 \cdot \sum_{i=1}^{C} p_i \cdot \sum_{i=1}^{C} r_i}{\sum_{i=1}^{C} p_i + \sum_{i=1}^{C} r_i} \tag{9}$$

Several further ideas about evaluation of classifiers are discussed in [32].

The results can be combined to *confusion matrices* which map real categories (listed in rows) to identified categories (listed in columns). The percentage numbers in the table cells correspond to the fraction how many music pieces are correctly classified or confused with other categories.

6 Summary and Outlook

Many recent publications address the field of music classification algorithms. In this paper, we have presented a wide overview of techniques without presenting the details. Several feature groups have been discussed, methods for feature pruning and classification strategies have been introduced. It is a snapshot of a continuously growing research field and should serve as a starting point to a literature study for the interested readers. Some observations and purposes can be made for further development:

- *Features*: The development of new features and intelligent methods of building meta-features can improve classification. Some feature domains are still difficult to investigate (tempo extraction, key and mode estimation) and further work is necessary.

- *Pruners* are not always widely used in MIR, new algorithms beyond statistical methods such as PCA and LDA are required. A promising idea is the application of EAs or the development of completely new techniques to measure the importance of music features.
- *Classifier* design belongs generally to the scope of data mining. New achievements in that area should be applied by the MIR community as well as the development of classification strategies with integrated and specific problem knowledge.

As a short and last conclusion, automatic music classification remains an exciting and growing research area, and many unsolved problems are still a challenge for the development of new algorithms.

7 Tools for Music Classification

7.1 jAudio

- *Application purpose*: Initially developed as part of ACE (Autonomous Classification Engine) framework and lately integrated in OMEN (On-demand Metadata Extraction Network) framework. jAudio is also available as stand-alone application
- *Developers*: Schulich School of Music, McGill University
- *Start of development*: 2005
- *Current status*: Beta release
- *Programming language*: Java
- *License*: LGPL
- *Supported audio formats*: Wave, mp3 and other formats supported by Java Sound API
- *Output formats*: Weka ARFF, ACE XML format
- *Batch and command line support*: Yes, batch files can be configured and saved for the future application with or without usage of GUI
- *Feature extraction details*: Currently over 20 distinct feature groups including signal and spectral properties, MFCC, beat histograms
- *Publications*: [39, 40]
- *WWW*: http://jaudio.sourceforge.net/

7.2 M2K (Music to Knowledge)

- *Application purpose*: Set of modules for D2K (Data to Knowlegde) framework, which should be used for various MIR tasks and their evaluation. Different extensions of M2K can be evaluated during the annual MIREX (Music Information Retrieval Evaluation eXchange) contests. Examples are provided of how to integrate experiment code from other environments such as C++ or Matlab.
- *Developers*: Graduate School of Library & Information Science, University of Illinois at Urbana-Champaign, The Automated Learning Group, The National Center for Supercomputing Applications, School Computing Sciences, University of East Anglia, Sun Microsystems Laboratories
- *Start of development*: 2004
- *Current status*: 1.2 release promised soon (incorporating evaluation code from MIREX 2006)
- *Programming language*: Java
- *License*: Academic use, research use and commercial evaluation licenses are available for D2K; M2K is distributed under free license
- *Supported audio formats*: Wave, mp3
- *Output formats*: D2K Table, ASCII file, ARFF (via D2K Table), Java serialization
- *Batch and command line support*: Yes
- *Feature extraction details*: Examples of various features are provided including MFCCs, spectral contrast features, common spectral shape descriptors (centroid, flux etc.) and onset detection functions
- *Classification algorithms details*: Classification modules are integrated in D2K and contain many strategies: decision trees, Bayesian classifier, Weka learners, neural networks, SVMs etc.
- *Publications*: [41]
- *WWW*: http://www.music-ir.org/evaluation/m2k/release/

7.3 MusicMiner

- *Application purpose*: Music browser which visualizes the differences between songs and artists
- *Developers*: Databionics Research Group, Universität Marburg
- *Start of development*: 2005
- *Current status*: Stable release
- *Programming language*: Java
- *License*: GPL
- *Supported audio formats*: Wave, mp3, ogg, wma, mp2, m4a
- *Output formats*: ASCII file
- *Batch and command line support*: Yes, feature extraction can be made from the command line
- *Feature extraction details*: Features are extracted using Yale ValueSeries preprocessing plugin (see below)
- *Classification algorithms details*: For the visualization of music similarities, emergent self-organizing maps are used
- *Publications*: [10, 42]
- *WWW*: http://musicminer.sourceforge.net/

7.4 RapidMiner (Yale)

- *Application purpose*: Environment for machine learning experiments and data mining. Feature extraction available via ValueSeries preprocessing plugin, numerous classification techniques are implemented
- *Developers*: Artificial Intelligence Unit, Universität Dortmund
- *Start of development*: 2001
- *Current status*: Stable release
- *Programming language*: Java
- *License*: GPL
- *Supported audio formats*: Wave, mp3, ogg
- *Output formats*: Weka ARFF and numerous other formats, which can also be user-defined
- *Batch and command line support*: Yes, batch XML files can be saved for future applications with or without usage of GUI
- *Feature extraction*: ValueSeries preprocessing plugin allows extraction of certain features (currently over 30 operators) as well as many transforms such as FFT, autocorrelation, phase domain transform, different filters etc.
- *Classification algorithms details*: Various classification techniques are available, including Weka learners, SVMs, Bayesian classifier, decision tree, association rules learners
- *Publications*: [43]
- *WWW*: http://rapid-i.com/

Acknowledgments

The authors would like to thank Dr. Martin Botteck (Nokia Research Center), Prof. Dr. Rudolph and Prof. Dr. Rötter (both from Universität Dortmund) for various discussions from engineering, computer science and music perspective, which helped to define the research topics and to review the existing literature. Also the authors thank the tool developers Daniel McEnnis (jAudio), Ingo Mierswa (Yale) and Kris West (M2K) for the quick answers and helpfulness during tool tests.

References

[1] F. Pachet and D. Cazaly, *A taxonomy of musical genres*, in Proc. Content- Based Multimedia Information Access (RIAO), Paris, France, 2000

[2] R. Typke, F. Wiering, and R. C. Veltkamp, *A Survey of Music Information Retrieval Systems*, in Proc. of the 6th International Conference on Music Information Retrieval (ISMIR), 2005, pp. 153-160

[3] I. Mierswa and K. Morik, *Automatic Feature Extraction for Classifying Audio Data*, Machine Learning Journal, 2005, vol. 58, pp. 127-149

[4] I. Mierswa, *Automatisierte Merkmalextraktion aus Audiodaten*, Mastersthesis, Universität Dortmund, Germany, 2004

[5] G. Tzanetakis and P. Cook, *Musical Genre Classification of Audio Signals*, IEEE Transactions on Speech and Audio Processing, 2002, vol. 10, pp. 293-302

[6] G. Peeters, *A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project*, IRCAM, France, 2004

[7] H. Pollard and E. Jansson, *A Tristimulus Method for the Specification of Musical Timbre*, Acustica, 1982, vol. 51, pp. 162-171

[8] A. Livshin and X. Rodet, *The Significance of the Non-Harmonic 'Noise' Versus the Harmonic Series for Musical Instrument Recognition*, in Proc. of the 7th International Conference on Music Information Retrieval (ISMIR), Victoria, 2006, pp. 95-100

[9] M. J. Hunt, M. Lennig and P. Mermelstein, *Experiments in Syllable-based Recognition of Continuous Speech*, in Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP) 1980, Denver, 1980, pp. 880-883

[10] F. Mörchen, A. Ultsch, M. Thies, I. Löhken, M. Nöcker, C. Stamm, N. Efthymiou, and M. Kümmerer, *MusicMiner: Visualizing Timbre Distances of Music as Topographical Maps*, Technical Report No. 47, Philipps-Universität Marburg, Germany, 2005

[11] N. Scaringella, G. Zoia, and D. Mlynek, *Automatic Genre Classification of Music Content*, IEEE Signal Processing Magazine, 2006, vol. 23, pp. 133-141

[12] M. Goto, *A Chorus-Section Detecting Method for Musical Audio Signals*, in Proc. ICASSP, 2003, pp. 437-440

[13] C. Krumhansl, *Cognitive Foundations of Musical Pitch*, Oxford University Press, New York, 1990

[14] D. Temperley, *The Cognition of Basic Musical Structures*, MIT Press, Cambridge, 2001

[15] G. Peeters, *Chroma-based estimation of musical key from audio-signal analysis*, in Proc. of the 7th International Conference on Music Information Retrieval (ISMIR), 2006, pp. 115-120

[16] Ö. Izmirli, *Audio Key Finding Using Low-Dimensional Spaces*, in Proc. of the 7th International Conference on Music Information Retrieval (ISMIR), 2006, pp. 127-132

[17] D. Butler, *The Musician's Guide to Perception and Cognition*, Schirmer Books, New York, 1992

[18] J. Seppännen, A. Eronen and J. Hiipakka, *Joint Beat and Tatum Tracking from Music Signals*, in Proc. of the 7th International Conference on Music Information Retrieval (ISMIR), Victoria, 2006, pp. 23 - 28

[19] N. Whitely, A. Cemgil and S. Godsill, *Bayesian Modelling of Temporal Structure in Musical Audio*, in Proc. of the 7th International Conference on Music Information Retrieval (ISMIR), Victoria, 2006, pp. 29 - 34

[20] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, New York, 1991

[21] L. R. Rabiner and B. H. Juang, *An Introduction to Hidden Markov Models*, IEEE Acoustic, Speech and Signal Processing Magazine, Vol. 3, 1986, pp. 4-16

[22] J. Arenas-García, J. Larsen, L. K. Hansen, and A. Meng, *Optimal filtering of dynamics in short-time features for music organization*, in Proc. of the 7th International Conference on Music Information Retrieval (ISMIR), 2006, pp. 290-295

[23] W. Chai, *Semantic Segmentation and Summarization of Music*, IEEE Signal Processing Magazine, 2006, vol. 23, pp. 124-132

[24] B. Whitman, *Learning the Meaning of Music*, Phdthesis, Massachusetts Institute of Technology, 2005

[25] L. I. Smith, *A tutorial on Principal Components Analysis*, 2002

[26] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, Wiley Interscience, 1992

[27] M. A. Hall, *Correlation-based feature selection machine learning*, PhD thesis, University of Waikato, New Zealand, 1998

[28] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor: The University of Michigan Press, 1975

[29] C.McKay and I.Fujinaga, *Musical genre classification: Is it worth pursuing and how can it be improved?*

[30] T. Kohonen, *Self-organizing Maps*, Springer, 1995

[31] F. Vignoli and S. Pauws, *A Music Retrieval System Based on User Driven Similarity and its Evaluation*, in Proc. of the 6th International Conference on Music Information Retrieval (ISMIR), 2005, pp. 272-279

[32] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2005

[33] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, 1993

[34] C. J. C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery 2(2), 1998, pp. 121-167

[35] J. Marques and P. Moreno, *A Study of Musical Instrument Classification Using Gaussian Mixture Models and Support Vector Machines*, Cambridge Research Laboratory, 1999

[36] A. Flexer, F. Gouyon, S. Dixon and G. Widmer, *Probabilistic Combination of Features for Music Classification*, in Proc. of the 7th International Conference on Music Information Retrieval (ISMIR), 2006, pp. 111-114

[37] T. Lidy and A. Rauber, *Evaluation of Feature Extractors and Psycho-Acoustic Transformations for Music Genre Classification*, in Proc. of the 6th International Conference on Music Information Retrieval (ISMIR), 2005, pp. 34-41

[38] J. Reunanen, *Overfitting in Making Comparisons Between Variable Selection Methods*, Journal of Machine Learning Research, Vol. 3, 2003, pp. 1371-1382

[39] D. McEnnis, C. McKay, I. Fujinaga and P. Depalle, *jAudio: A Feature Extraction Library*, in Proc. of the 6th International Conference on Music Information Retrieval (ISMIR), 2005, pp. 600-603

[40] D. McEnnis, C. McKay, I. Fujinaga and P. Depalle, *jAudio: Additions and Improvements*, in Proc. of the 7th International Conference on Music Information Retrieval (ISMIR), 2006, pp. 385-386

[41] J. S. Downie, J. Futrelle, D. Tcheng, *The International Music Information Retrieval Systems Evaluation Laboratory: Governance, Access and Security*, in Proc. of the 5th International Conference on Music Information Retrieval (ISMIR), Barcelona, 2004

[42] A. Ultsch and F. Mörchen, *ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM*, Technical Report No. 46, Philipps-Universität Marburg, Germany, 2005

[43] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler, *YALE: Rapid Prototyping for Complex Data Mining Tasks*, in Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM Press, 2006

