# Audio Feature Engineering for Automatic Music Genre Classification

**Paolo Annesi, Roberto Basili**
**Raffaele Gitto, Alessandro Moschitti**

Department of Computer Science, Systems and Production,
University of Roma, Tor Vergata, ITALY
{basili, moschitti}@info.uniroma2.it, {paolo.annesi, r.gitto}@gmail.com

**Riccardo Petitti**

Exprivia S.p.A. Via Cristoforo Colombo 456,
00145 Roma, ITALY
riccardo.petitti@exprivia.it

**Abstract**

The scenarios opened by the increasing availability, sharing and dissemination of music across the Web is pushing for fast, effective and abstract ways of organizing and retrieving music material. Automatic classification is a central activity to model most of these processes, thus its design plays a relevant role in advanced Music Information Retrieval. In this paper, we adopted a state-of-the-art machine learning algorithm, i.e. Support Vector Machines, to design an automatic classifier of music genres. In order to optimize classification accuracy, we implemented some already proposed features and engineered new ones to capture aspects of songs that have been neglected in previous studies. The classification results on two datasets suggest that our model based on very simple features reaches the state-of-art accuracy (on the ISMIR dataset) and very high performance on a music corpus collected locally.

# 1 Introduction

Music genres are difficult to describe as there is no complete agreement on their definition. "*Genres emerge as terms and nouns that define recurrences and similarities that members of a community make pertinent to identify musical events*" (Fabri, 1997)
The notion of community corresponds to a complex self-organizing system that triggers the development and assessment of a *genre*. In this perspective, the community plays the role of an ontology designer which implicitly defines properties and rules of the target genre as well as its differences with external habits and trends.

Given the high complexity of such system, to define a model for automatic genre classification, we should capitalize from the work carried out in Information Retrieval (IR). This has shown that document relevance with respect to a user's query (e.g. a particular song) is not determined by only local properties, e.g. the query and the retrieved items, as global notions, that emerge from the entire corpus, are also important. Indeed, every quantitative model in IR relies on a large number of parameters (e.g. *term weights*) that depend on the set of *all* indexed documents. In order to model a musical genre, it is

thus critical to study local (the target genre examples) and global (the examples of other genres) characteristics and express them in term of statistical properties.

Such concepts are the foundations of modern machine learning algorithms (Mitchell, 1997) which aim to model classification functions based on the sets of positive and negative examples, i.e. the songs that belong or not to a target genre. As the machine learning approaches are quite standard and they tend to behave similarly on different application domains, the actual complexity relates mainly to the feature design task. The role of features is to provide a description of example songs that can be processed by learning algorithms. These will guide the induction of the classification function in agreement with such descriptions. As we would like to classify songs stored as audio files, i.e. waveforms, the design of features is quite complex and requires the application of signal analysis techniques.

In this paper, we experimented a state-of-the-art machine learning algorithm, i.e. Support Vector Machines, in the design of an automatic genre classifier over audio information. In order to optimize the classification accuracy of our model, we implemented some features described in literature and designed new features to capture aspects previously neglected. We experimented our models on annotated collections (i.e. classified data instances) made available in previous investigation (*Magnatune* dataset) as well as on a novel data collection, designed to carry out a cross-collection comparison. The results obtained on large scale experiments suggest that our model based on very simple features reaches the state-of-art accuracy on the *Magnatune* dataset and very high performance on our new music corpus.

The remainder of this paper is organized as follows: Section 2 introduces Support Vector Machine and kernel methods, Section 3 describes the basic and new set of features, Section 4 shows the experiments with music genre categorization and finally Section 5 summarizes the conclusions.

## 1.1 Related Work

The description of the basic features used in our experiments can be found in (Tzanetakis et al., 2001). The main idea that we inherited from Tzanetakis et al. is the split between superficial features, called *musical surface* and advanced features, i.e. *rhythm features*. The literature experiments show that such baseline features achieve very high accuracy. Another very inspiring work is (Tzanetakis et al., 2002) in which the concept of beat strength is defined as a rhythmic characteristic that allows us to discriminate between two pieces having the same tempo.

The *MFCC* feature has been used in music categorization in (Logan, 2000). Such study demonstrates the importance of using *MFCC* for music classification. Moreover, MFCC was used in the Pampalk's system that won the MIREX 2004 competition. In such work, the feature extraction method was based on a frame cluster similarity (Pampalk, 2005). *MFCC* was also used in another system based on AdaBoost (Bergstra and Casagrande, 2005) that won MIREX 2005.

In (Lidy and Rauber, 2005), a combination of features based on rhythm patterns, statistical descriptors and rhythm histograms was used. In (Gouyon et al., 2004), it was considered a specific set of rhythmic descriptors for which was provided procedures of automatic extraction from audio signals. The corpus used in our experimentation was also used in (Pampalk et al., 2005), with a set of *spectral features*, e.g. *MFCC*, and a set of advanced features, called *fluctuation patterns*.

In (Berenzweig et al., 2003), it is described a method of music mapping into a semantic space that can be used for music similarity measurement. The value along each dimension of this *anchor space* is computed as the output from a pattern classifier which is trained to measure a particular semantic feature. In anchor space, distributions that represent objects such as artists or songs are modeled with Gaussian Mixture Models. An interesting approach is used in (Mandel et al., 2005) where it is described a system for performing flexible music similarity queries using SVM active learning. In (Lippens et al., 2004) it is shown that, although there is room for improvement, genre classification is inherently subjective and therefore perfect results can not be expected neither from automatic nor from human classification.

# 2 Genre Classifier based on Support Vector Machines

Many learning algorithms consider features as dimensions of a vector space. Each instance is represented by a feature vector where the components are the numeric values associated with features. Support Vector Machines (SVMs) (Vapnik, 1995) are state-of-the-art learning methods based on vector spaces. One of their interesting properties is the possibilities of using kernel functions. These allow SVMs to implicitly generate large feature spaces like for example the space of feature conjunctions. The next section briefly introduces this interesting machine learning approach.

## 2.1 Support Vector Machines

To apply SVMs to music classification, we need a function $\phi : \mathcal{S} \rightarrow \Re^n$ to map our song space $\mathcal{S}$ into $\Re^n$. Given such vector space and a set of positive and negative examples mapped in vectors, SVMs classify them according to a separating hyperplane, $H(\vec{x}) = \vec{w} \cdot \vec{x} + b = 0$, where $\vec{x} = \phi(s), s \in \mathcal{S}$ and the two parameters $\vec{w} \in \Re^n$ and $b \in \Re$ (learned by applying the *Structural Risk Minimization principle* (Vapnik, 1995)). More in detail, they are learned by solving the following optimization problem:

$$\begin{cases} min \quad ||\vec{w}||^2 + C \sum_{i=1}^{m} \xi_i^2 \\ y_i(\vec{w} \cdot \vec{x_i} + b) \geq 1 - \xi_i, \quad \forall i = 1, .., m \\ \xi_i \geq 0, \quad i = 1, .., m \end{cases} \tag{1}$$

where $\vec{x_i}$ are the training instances, $m$ is the number of such instances and $\xi_i$ are the slack variables of the optimization problem. From the kernel theory we have that:

$$H(\vec{x}) = \Big( \sum_{i=1..l} y_i \alpha_i \vec{x_i} \Big) \cdot \vec{x} + b = \sum_{i=1..l} y_i \alpha_i \vec{x_i} \cdot \vec{x} + b = \sum_{i=1..l} y_i \alpha_i \phi(s_i) \cdot \phi(s) + b = 0.$$

where $y_i$ is equal to 1 for a positive example and or -1 for a negative example, $\alpha_i \in \Re$ with $\alpha_i \geq 0$, $\phi(s_i) = \vec{x_i}$ $\forall i \in \{1, .., l\}$ are the training instances and the product $K(s_i, s) = \langle \phi(s_i) \cdot \phi(s) \rangle$ is the kernel function associated with the mapping $\phi$. The simplest mapping that we can apply is $\phi(s) = \vec{x} = \langle x_1, ..., x_n \rangle$ where $x_i = 1$ *if* the feature $i$ appears in the song $s$ otherwise $x_i = 0$. If we use as a kernel function the scalar product, we obtain the linear kernel $K_L(s_i, s) = \vec{x_i} \cdot \vec{x}$.

Another interesting kernel is the polynomial one, i.e. $K_p(s_i, s) = (c + \vec{x_i} \cdot \vec{x})^d$, where $c$ is a constant and $d$ is the degree of the polynom (Basili and Moschitti, 2005). The polynomial kernel is equivalent to carry out the scalar product in the space of feature conjunctions, where the number of features in each conjunction is up to $d$. For example, if we have two features such as *pitch* and *volume*, the learning algorithm can test if some

combinations characterizes a particular genre, e.g. the combination of low *pitch* and low *volume* is typical of *classical music*. Although, kernel methods are a powerful tools to learn class differences, it is very important to define a *good* set of features achieving optimal results.

# 3 Extracting Features from Audio Files

In this paper, we use several basic features proposed in music classification literature and we also propose new interesting ones. The next sections are devoted to the description of the experimented features.

## 3.1 Simple or Basic Features

We represent the musical surface of each song by means of the statistics of the spectral distribution over time. In particular, we analyze the average and standard deviation of 6-dimensional vectors over the entire song. Such dimensions, *volume*, *beats*, *spectral energy*, *centroid*, *pitch* and *5-MFCC*, are described hereafter.

- *Volume*. Given $N$ song samples, $S_k$, the *Volume* is:

$$Volume = \sqrt{\frac{1}{N}\sum_{k=1}^{N} A(S_k)^2} \tag{2}$$

  where $S_k$ is one of the samples stored in the buffer and $A(S_k)$ is the amplitude of the signal at time $S_k$. This function is not equivalent to the concept of volume used in signal processing, but it gives higher values for louder sounds and viceversa, which is enough for our purposes. Moreover, during the preprocessing phase, we normalize the volume of each song as we are not interested to absolute values but to the relative differences between two frames.

- The *Spectral Energy* is correlated to the Fourier Transform, which maps audio signal into frequency domain. For each audio sample set, we compute Fast Fourier Transformation (FFT).

- *Centroid* is an interesting psycho-acoustical feature that measures the mean spectral frequency in relation with the amplitude; in other words the position in Hz of the center of mass of the spectrum. it is useful as a measure of the sound brightness.

$$C = \frac{\sum_{k=0}^{N/2} f_k |X(k)|}{\sum_{k=0}^{N/2} |X(k)|} \tag{3}$$

  where N is the FFT size, $X(k)$, $k = 0, .., N$ is the FFT of the input signal, and $f_k$, $k = 1, .., N$, is the k-th frequency bin.

- *Pitch* is the perceived fundamental frequency of a sound. This can be computed by an autocorrelation algorithm applied to audio signals. We defined a different approximation of the above notion as it captures more information. We define chroma vectors as 12-element vectors, where each component represents the spectral energy corresponding to one pitch class (i.e. C, C#, D, D#, etc.). The algorithm, builds the 12 chroma vectors by deriving components from the main frequencies

in a temporal frame. Durations below some tresholds are not taken into account as they are considered not meaningful for the frame. Notes corresponding to each frequency are then mapped according to a fuzzy matching based on a reference octave frequency: in order to discretize, i.e. select the proper note, the frequencies of an $\mathcal{A}$ note in every octave (for a total of 8 octaves) are taken as reference.

- The *Mel Frequency Cepstral Coefficients* (MFCCs) are well known compact forms that can represent speeches. They are the most common representation used for *Spectra* in Music Information Retrieval (MIR). The following is a brief algorithm for their computation:

  1. apply window function;
  2. compute power spectrum (using FFT);
  3. apply Mel filter bank;
  4. apply Discrete Cosine Transform (DCT);

  The MFCCs have important advantages: they are simple and fast, well tested. Moreover, they have also a compressed and flexible (i.e. easy to handle) representation (Logan, 2000).

## 3.2 Complex, Synchronous and Structural Features

In this section we describe our new designed features.

### 3.2.1 Beats

The *Beats* feature tries to count the number of beats of a song. Generally, beats are driven by instruments that operate in the lower frequencies, like the drum or the bass. In order to obtain a feature able to count the number of beats, we apply a lowpass to the target song. This will cut off frequencies higher than 200 Hz ((McNab et al., 1996), (Marolt, 2006), (Davies and Plumbley, 2005)). Most instruments playing in a song will be attenuated or totally eliminated. The remaining sounds are usually related to the drums and the bass.

More in detail, our algorithm uses the volume feature value together with 10 low-pass filters that cut over the frequencies higher than 200 Hz. We use a bank of ten filters in order to minimize the sound distorsion and the computation time. Then, the song wavelength is discretized to analyze the *attack* of each beat (referred as *peak duration* below), i.e. the time that elapses between the start of a peak and the successive silent phase. As the temporal range of the target frame is fixed the number of peaks is also informative about the rhythm (i.e. it is a crude but useful approximation of the notion of *bpm* local to a sample). An example of this process is given in Figure 1. Given such discretized wave, we extract five values:

- average and variance of peak duration
- average and variance of peak distance
- average beats per minute

The figure 2 shows two music moments that can clearly be distinguished using the beats feature. In the first moment, a series of regular impulses caused by a drum is present whereas in the second we find a more complex texture produced by a bass. Moreover, analyzing the amplitude of a wave, we can determine the classification of genres: songs of classical and jazz genres show lower waves contrarily to rock and electronic songs.

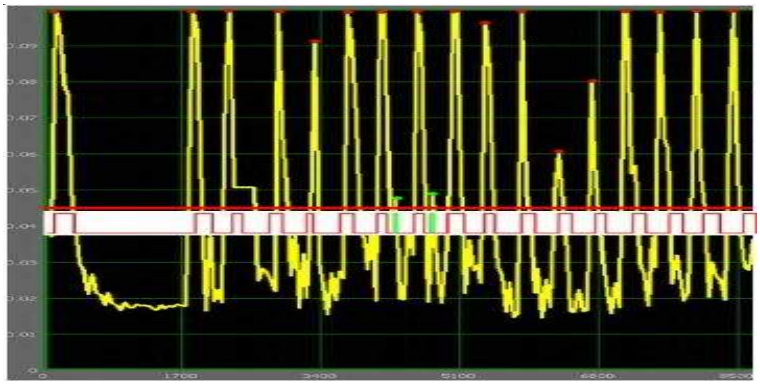

Figure 1: An example of beats for electronic music

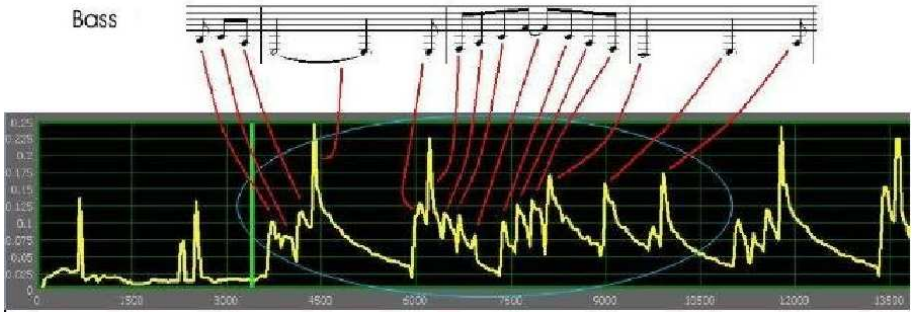


Figure 2: An example of beats for jazz-blues music

### 3.2.2 Volume Reverse

The intuition about this feature come from examining song recording methods. The first step in a recording process is to collect the sounds of every instrument. Several microphones can be used for the same instrument or, viceversa, the same microphone can be used for several instruments. When the recording phase of a single track is completed, a master multi-track is mixed in stereo channels so that a song can be played by conventional hi-fi equipments. For example, rock, pop and electronic music, is often produced by moving a sound of one instrument from one stereo channel to the other (sound effects like echo and surround). Moreover, such music is enriched with sonorous effects. For example, with rock music guitar distortions are often used to make the sound less uniform whereas with pop and electronic music scratch effect is applied.

The above techniques make the audio wave of a channel very different from the other. On the contrary, classic and jazz music is recorded with different modalities. First of all, instruments' distortions and sonorous effects are quite rare because this music is based on a cleaner type of sound. The recording technique is direct and makes a large use of environment microphones as it is preferred to emphasize live recording, giving much more importance to the solos and improvisations. This produces a stereo track with two very similar channels.

By considering the different recording methods, we can distinguish rock, pop and electronic music from classic and jazz. For this purpose, we designed a feature that measures the variation between the sound wave of the two stereo channels. More in detail, we subtract the audio wave of a channel to the other one and compute the absolute value. We expect that for classic and jazz music the values tend to be around zero whereas for

| **Features** | *Multiclassifier* | *Rock* | *Classic* | *Jazz* | *Electronic* | *Metal* | *World* |
|---|---|---|---|---|---|---|---|
| | *Accuracy* | F1 measure | | | | | |
| Basic | 80.5 | 0.63 | 0.93 | 0.68 | 0.76 | 0.59 | 0.73 |
| Basic + Pitch | 80.0 | 0.61 | 0.93 | 0.65 | 0.76 | 0.56 | 0.72 |
| Basic + Beats | 80.4 | 0.64 | 0.93 | 0.6 | 0.8 | 0.57 | 0.7 |
| Basic + Chorus | 80.6 | 0.62 | 0.93 | 0.68 | 0.77 | 0.58 | 0.73 |
| Basic + Reverse | 80.9 | 0.62 | 0.94 | 0.72 | 0.77 | 0.58 | 0.74 |
| Basic + Reverse + Chorus | 82.0 | 0.66 | 0.94 | 0.7 | 0.79 | 0.61 | 0.74 |
| Basic + Reverse + Chorus + Pitch | 81.9 | 0.64 | 0.94 | 0.7 | 0.80 | 0.58 | 0.74 |
| Basic + Reverse + Chorus + Beats | 81.2 | 0.62 | 0.93 | 0.68 | 0.82 | 0.58 | 0.71 |
| Basic + All Advanced | 82.3 | 0.66 | 0.94 | 0.68 | 0.83 | 0.61 | 0.72 |
| Pampalk 2004 | 84,10 | - | - | - | - | - | - |

Table 1: Accuracy on the Magnatune 2004 Corpus

| **Features** | *Multiclassifier* | *Rock* | *Classic* | *Jazz* | *Electronic* | *Pop* |
|---|---|---|---|---|---|---|
| | *Accuracy* | F1 measure | | | | |
| Basic | 89.4 | 0.85 | 0.93 | 0.9 | 0.91 | 0.87 |
| Basic + Pitch | 90.8 | 0.89 | 0.93 | 0.9 | 0.91 | 0.9 |
| Basic + Beats | 89.8 | 0.85 | 0.94 | 0.9 | 0.92 | 0.86 |
| Basic + Chorus | 88.4 | 0.83 | 0.93 | 0.9 | 0.91 | 0.85 |
| Basic + Reverse | 90.0 | 0.86 | 0.94 | 0.91 | 0.91 | 0.88 |
| Basic + Reverse + Chorus | 90.8 | 0.87 | 0.94 | 0.91 | 0.92 | 0.89 |
| Basic + Reverse + Chorus + Pitch | 91.6 | 0.9 | 0.94 | 0.9 | 0.92 | 0.92 |
| Basic + Reverse + Chorus + Beats | 91.4 | 0.88 | 0.95 | 0.93 | 0.91 | 0.89 |
| Basic + Advanced (ALL) | 92.0 | 0.89 | 0.96 | 0.92 | 0.91 | 0.91 |

Table 2: Accuracy on the RTV Corpus

elettronic, rock and pop values are subject to sudden changes.

### 3.2.3 Chorus

A characteristic of rock and pop songs is the presence of a periodic structure: to a verse follows a chorus and so on until the end of the song. It is also applied a change of tonality at the end of the song. This schema is less strict in the electronic, jazz and classics songs. The latter two musical genres show more improvisation and the presence of very technical solos which make songs much more complex and less rigid in their internal structure. Finding a general schema of the songs can help to distinguish between jazz, classic and electronic genre from music much closer to rock and pop.

Our algorithm to detect such schema eliminates the voice of the singer (if there is any), to preserve only the audio data given by musical instruments. This is carried out by subtracting the wave of the two audio channels (of course, this can be done only if the analyzed song is stereo). Such approach will eliminate the middle channel audio which generally is the part containing the singer voice.

After the above step, a schema of the song can be detected based on the analysis of the spectral energy. In particular, we have noted that the chorus of pop or rock music is associated with greater values of energy. Thus, our algorithm computes mean and mean square values of the audio wave of the frequency of the detected chorus part.

# 4 Experiments

In these experiments, we tested our SVM song classifier on two different data sets, *Magnatune 2004* and a novel music corpus (RTV) that contains some *Magnatune* songs mixed with commercial music. We experimented with different feature combinations to study

their usefulness in characterizing different genres.

## 4.1 Experimental Setting

The *Magnatune 2004* corpus is composed by 729 songs distributed in 6 genres as follows *Rock* 13.7%, *Classic* 44.2%, *Jazz* 3.6%, *Electronic* 15.8%, *Metal* 6.2% and *World* 16.6%. All the songs are free from copyrights and can be downloaded from `http://ismir2004.ismir.net/`. This corpus is quite difficult to classify for at least three reasons: (1) the *World* genre classification is quite complex as it encloses several different sounds and styles; (2) there is a strong similarity between music *Rock* and *Metal*; and (3) there are very few examples of *Jazz*.

The RTV corpus is composed by 500 songs selected from *Magnatune* and some songs selected by proprietary databases[1]. Such songs are equally distributed on 5 genres (each of them contains 100 songs): *Rock*, *Classic*, *Jazz-blues*, *Electronic* and *Pop*. *Rock* and *Pop* classes are composed by commercial music (e.g. Madonna and Depeche Mode for Pop and Metallica and Korn for Rock). The songs of the other classes are randomly selected from *Magnatune*.

For the experiments, we used the WEKA software available at `http://www.cs.waikato.ac.nz/~ml/weka/`. We used the default parameters and the polynomial kernel (of Waikato, 2006). We trained an SVM for each class in the scheme ONE-vs-ALL (Rifkin and Klautau, 2004). For each testing instance, we selected the class associated with the highest SVM score. The classification performance of the individual class is evaluated with the $F_1$ measure. This assigns equal importance to Precision $P$ and Recall $R$, i.e. $f_1 = \frac{2P \times R}{P+R}$. The multiclassifier performance is measured by means of accuracy.

## 4.2 Classification Results

For both corpora we applied a 10-fold cross validation. This means that we divided each corpus in 10 parts and 9 of them were used for training and 1 for testing. By rotating the testing sample, we obtained 10 different measures on which we evaluated the average.

Table 1 reports the results for the *Magnatune* collection. Column 1 shows the feature sets used to represent the song instances, Column 2 reports the multiclassifier accuracy, and the columns from 3 to 6 illustrates the $F_1$ measures of the *Rock*, *Classic*, *Jazz*, *Electronic*, *Metal* and *World* binary classifiers, respectively. We note that the more features we use the higher the multiclassifier accuracy is. Indeed, the best result is achieved using the basic features plus the new ones, i.e. 82.3%. Consequently, the new features improve the basic features of about 2 absolute percent points. This enhancement is not neglectable since it is difficult to improve an already high baseline, i.e. 80.45%. Moreover, 82.3% is very near to the best figure obtained on the *Magnatune* corpus, i.e. the 84.1% derived in (Pampalk, 2005). Note that the features used to obtain such state-of-the-art accuracy are remarkably more complex to extract than those proposed in our model. Such complexity made difficult to implement and study a model that combines such features with those that we propose. Although this will be part of our future work.

Regarding the individual categories, we observed from the confusion matrix that *Rock* is often misclassified in place of *Metal* and viceversa. The *Jazz* classifier has a low accuracy; this suggests that it is difficult to recognize *Jazz* songs. An alternative explanation is

[1]As these songs are protected by copyrights, we could not make RTV available but we are going to provide the learning files in WEKA format.

the low number of training instances available to train the corresponding binary classifier. On the contrary, the accuracy on *Classic* and *Electronic* genres is quite high. This can be explained by the remarkable differences in terms of musicological and sonorous aspects.

With the aim of showing that our features capture important difference between the diverse genres, we experimented our classifiers on the RTV collection. The results are reported in Table 2, which is very similar to the previous Table except for the presence of the Pop category. We note that the accuracy is in general much higher than the one obtained on the *Magnatune* test set. The main reason is that in RTV each category has an enough number of positive examples for training (i.e. 100). This does not happen for *Magnatune*. For example, *Jazz* has only 26 training songs. Moreover, we still observe an improvement of about 2% of the classification accuracy when the new features are added to the basic ones.

In particular we empathize the relevance of the feature *Volume Reverse* in the classification results related to the *jazz* and *world*. When this feature is added to the set the f-measure of this genre reaches the peak.

Finally, it should be noted that also the accuracy obtained with the basic features is very high on both collections. This is due to the use of (a) a powerful learning algorithm, i.e. SVMs, and (b) the polynomial kernel that generates many interesting feature conjunctions.

# 5 Conclusions

The large availability of songs across the Web requires effective ways of automatically organizing and retrieving music material. Automated genre classification is thus a critical step to carry out such processes.

In this paper, we adopted a state-of-the-art machine learning algorithm, i.e. Support Vector Machines, to design an automatic classifier of music genres. To improve the classification accuracy of our system, we used previous designed features and we engineered new ones. The classification accuracy on two datasets show that our model based on very simple features approaches the state-of-art systems. This good result is due to both our novel features and the use of a powerful learning model, i.e. SVMs along with the very promising techniques based on kernel methods.

# References


R. Basili and A. Moschitti. *Automatic Text Categorization: from Information Retrieval to Support Vector Learning*. Aracne Publisher, 2005.

A. Berenzweig, D. Ellis, and S. Lawrence. Anchor space for classification and similarity measurement of music. In *Proceedings of IEEE ICME*, 2003.

J. Bergstra and N. Casagrande. Two algorithms for timbre and rhythm-based multiresolution audio classification. In *Proceedings of ISMIR*, 2005.

M. E. P. Davies and M. D. Plumbley. Beat tracking with a two state model. In *Proceedings of IEEE ICASSP*, 2005.

F. Fabri. Browsing music spaces: Categories and the musical mind. Website, 1997. URL `http://www.mediamusicstudies.net/tagg/others/ffabbri9907.html`.

F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer. Evaluating rhythmic descriptors for musical genre classification. In *Proceedings of AES 25th International Conference*, 2004.

T. Lidy and A. Rauber. Combined fluctuation features for music genre classification. In *Proceedings of MIREX*, 2005.

S. Lippens, J.P. Martens, T. De Mulder, and G. Tzanetakis. A comparison of human and automatic musical genre classification. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2004.

B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of ISMIR*, 2000.

M. Mandel, G. Poliner, and D. Ellis. Support vector machine active learning for music retrieval. *ACM Multimedia System Journal*, 2005.

M. Marolt. A mid-level melody-based representation for calculating audio similarity. In *Proceedings of ISMIR*, 2006.

R.J. McNab, L.A. Smith, I.H. Witten, C.L. Henderson, and S.J. Cunningham. Towards the digital music library: Tune retrieval from acoustic input. In *Proceedings of the ACM Conference on Digital Libraries*, 1996.

T. M. Mitchell. *Machine Learning*. Mc Graw Hill, 1997.

University of Waikato. Weka, a platform for automatic classification. http://www.cs.waikato.ac.nz/ ml/weka/. Website, 2006. URL `http://www.cs.waikato.ac.nz/ ml/weka/`.

E. Pampalk. Speeding up music similarity. In *Proceedings of MIREX*, 2005.

E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classificaton. In *Proceedings of ISMIR*, 2005.

R. Rifkin and A. Klautau. In defense of one-vs-all classification. *J. Mach. Learn. Res.*, 5: 101–141, 2004. ISSN 1533-7928.

G. Tzanetakis, G. Essl, and P. Cook. Automatic musical genre classification of audio signals. In *Proceedings of ISMIR*, 2001. URL `gtzan@cs.princeton.edu`.

G. Tzanetakis, G. Essl, and P. Cook. Human perception and computer extraction of musical beat strength. In *Procedings of Digital Audio Effects (DAF)*, 2002.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.