Musical genre classification

ROBERT NEUMAYER,
a0003208@unet.univie.ac.at

Abstract. Musical genres are labels used to distinguish between different types or categories of musical style. The growing amount of music that is available electronically, particularly in the World Wide Web, creates an arising need for automated classification. A typical application could be plugins for music management software or some sort of tool that automatically tags audio files according to its genre. This article covers the possibilities of musical genre classification and its application to mp3-files. Mainly it deals with the classification accuracies obtained by the use of different music collections using a multi layer perceptron. Moreover it discusses the similarities or differences between different genres according to a specific set of features.

1 Introduction

In particular the lots of music that is available via internet shows the growing demand of automatic classification in order to categorize and index those media files. This work could contribute to, or support any application that automatically assigns genres to audio files. This article will mainly concentrate on how to classify any pieces of music according to derived feature vectors. The main goal is to imitate human classifiers who assign a certain genre according to their listening impression.

1.1 Musical Genres

Humans label audio files according to different criteria heavily influenced by personal preferences. There is no strict boundaries between different genres, nor a significant coincidence in classification results of different people can be observed (many people would label a given piece of music as one genre, whereas many others would label it completely different). Classification is easier on datasets with labels that anyone would agree to like colors, etc.. So the work presented in this article mainly is about training a system to imitate my own classification decisions (that I made when assigning the ID3-Tags to the mp3-files).

1.2 Related Work

Many people were dealing with the problem of feature extraction or music and speech discrimination. Tzanetakis and Cook have done a lot of work in this field, one of their articles will be some kind of reference for this one [3]. In particular

the development of Tzanetakis' MARSYAS [7], which is a framework for both feature extraction from and classification of audio files, is an important prerequisite for this article. The structuring will be as follows: Section (2) introduces common techniques used of audio feature extraction and its application to classifying genres. Section (3) discusses the use of multi layer perceptrons for classification purposes and Section (4) reviews the application of those techniques by experiments with a sample music collection. The results will be compared to sort of reference results in Section (5). Section (6) presents a short conclusion and some ideas not covered by this article.

2 Input Data And Feature Extraction

2.1 Audio Feature Extraction

The main focus of this article is not feature extraction itself. I use MARSYAS for the extraction of the features which generates a vector of 30 features for each music sample. MARSYAS uses three different types of features:

- Timbral Features
- Rhythmic Content Features
- Pitch Content Features

Their combination should provide for a quite good discrimination of musical genres.

2.2 Getting the input and target data

As the goal is to classify mp3-files according to their genre, and classification is an application of supervised learning, there is a need for targets. Those vectors were constructed by reading out the genre of each mp3 and setting the corresponding flag of the vector to one. (e.g. four genres in the collection: Pop, Heavy Metal, Punk Rock, Hip-Hop, the target row for any song belonging to the Hip-Hop genre would be: 0 0 0 1). Classification in neural networks involves gathering the input data (the audio features) and output data (a genre information) for each audio file. So each file of an existing collection will be used to process a feature vector (4,0.4,0.9,59,30) and a target/genre vector (0,0,0,1,0,0). As the existing collection is in mp3-format and every mp3-file has an assigned genre (that could eventually be an empty string or unknown - those files of course can not be taken into account). The target value for each file is read out from the ID3 Tag of the (mp3) file. Actually this is the connection to imitate the human classifier, because those genre tags are a sort of subjective labeling and represent personal preferences. Those feature- and genre vectors should then be the input respectively target vectors for a system to propose a genre to each file. A multi layer perceptron will be used as classification mechanism in this case.

3 Neural Network Based Classification

I am using the Netlab Toolbox of Aston University [1]. Classification is done through multi layer perceptrons. For an existing collection two matrices, containing feature vectors and target vectors, are calculated. Those data is still *raw* data and needs some preprocessing. Any type of input data has to be normalized. The input and target matrices have to be randomized (synchronized of course). Another very important point is splitting up into training and test data, which in this case is done by splitting up each input and target vector to 85 per cent training and 15 per cent testing data. Furthermore every training set is split into n pieces of equal size, *n-1* for training and one for testing. This results in n nets classifying the same problem but based on different training and validation sets whose classification accuracy is averaged. Cross-validation produces results that are far less dependent on the specific training set than the simple approach including only one training and testing set. In the end the nets get tested with the previously splitted testing set. Actually this leads to very similar results, because the validation data is not completely independent from the training set (which is an important fact, when trying to get an impression of the classification quality for new datasets). The dependence between testing and training set is given because I did not differentiate between artists or albums, only between songs. So it is not possible for one song to be part of both training and evaluation set, but it is possible to songs from the same artist and album.

4 Experiments

4.1 A General Impression of the Music Collection

The collection available for experiments consists of about 3500 mp3-files. Table (1) shows the initial composition of the available music collection. It consists of 34 different genres and 3445 tracks, whereas many of them are hardly usable. There are many tracks without a genre assignment that wouldn't advance proper classification because of its distorting impact. There are many more indicators of the poor data quality like the genre Comedy that would rather belong to speech etc..

4.2 A First Shot

In spite of the fact that this is probably an above average collection concerning data quality[1] the collection is far from perfect, but should be enough to gain experience in possibilities and limitations of classifying genres. To get a general impression about how it would work, I reduced the number of genres and removed or corrected any genre assignments that were not appropriate or not existent. This leads to the collection presented in Table (2).

[1] in terms of only complete albums, (almost) all ID3-Tags assigned correctly (in my opinion), ...

Genres	Number of tracks
Punk Rock	775
Pop	349
Hip-Hop	306
Rock	237
Metal	209
Hardcore	140
NO GENRE	137
Punk	133
Alt. Rock	120
R&B	115
Alternative	111
Grunge	103
Slow Rock	83
Industrial	77
Ska	68
Avantgarde	60
Rock & Roll	53
Reggae	50
Acid Punk	46
Noise	45
BritPop	35
Death Metal	31
Other	21
Classical	17
Country	15
Indie	13
Club	13
Electronic	13
Hard Rock	11
Comedy	10
Instrumental	9
Duet	5
Dance	2
Number of genres: 34	Number of tracks: 3445

Table 1. Overview of available genres and the number of tracks associated to that genre

Genres	Number of tracks
Punk	695
Pop	269
Hip-Hop	236
Rock	205
Metal	161
no genre assigned	158
Alt. Rock	120
Alternative	105
Hardcore	80
R&B	57
Rock & Roll	53
Grunge	47
BritPop	35
Avantgarde	31
Ska	31
Noise	29
Acid Punk	23
Soundtrack	18
Country	15
Industrial	13
Instrumental	9
Number of genres: 21	Number of tracks: 2390

Table 2. Overview of the cleansed music collection

Classification of this collection resulted in a classification accuracy of about 20 per cent using a linear perceptron and about 37 per cent using a multi layer perceptron[2]. Although those results are far from randomly guessing which would be 1 divided by 20 so 0.05, they were not very impressive. The poor accuracy could either indicate poor quality of the features or simply an under-determined network, what shall be examined in the upcoming sections.

4.3 Getting Closer

Further I tried to take very few (four in this case) genres that were quite different from each other as shown in Table (3).

Genres	Number of tracks
Metal	94
Punk Rock	88
Grunge	47
Hip-Hop	99
Number of genres: 4	Number of tracks: 328

Table 3. Overview of the shrinked collection

The classification accuracy rose to about 81 per cent using a linear perceptron and 72 per cent using a multi layer perceptron[3]. Note that the linear perceptron is superior in this case. This result indicates that rather good classification accuracy is possible although a classification in a set of four different genres may seem a bit weak or useless. Therefore I tried to get a bigger collection that delivers a usable classification accuracy despite of its greater size.

4.4 Finally, a Nice T estcollection

I decided to perform any upcoming experiments on the music collection presented in Table (4). This collection contains the genres with the most samples in my collections (e.g. did not include the genre Noise because there's only 50 tracks). The Classical genre consists of actually two genres: opera and classical (it's rather the same in my opinion), the Hip-Hop genre actually consists of Hip-Hop and R&B. Hardcore is more or less a mix of Hardcore and Emo genres. Table (5) should give you an overall impression of the associated artists (I know it's a great reason for dispute ...). This (Figure (4)) should be an acceptable music collection. The model won't be under-determined up to about 4 hidden

[2] 150 cycles, fifteen-fold cross-validation, 20 hidden units
[3] 150 cycles, fifteen-fold cross-validation, 3 hidden units

Genres	Number of tracks
Reggae	99
Classical	277
Metal	190
Hardcore	127
Slow Rock	96
Punk Rock	171
Grunge	119
Hip-Hop	336
Pop	240
Number of genres: 9	Number of tracks: 1655

Table 4. A practical testcollection

Genres	Important Artists
Reggae	Bob Marley, ...
Classical	Wagner, Beethoven, Mozart, Vivaldi, Schubert
Metal	Cult Of Luna, Napalm Death, Soulfly, Meshuggah, Il Nino, ...
Hardcore	The Suicide Machines, Poison The Well, Red Tape, ...
Slow Rock	Radiohead, Lambchop, Snow Patrol, Kashmir, Aqualung, The Crash, ...
Punk Rock	NOFX, Pennywise, No Use For A Name, Lagwagon, Refused, ...
Grunge	Nirvana, Melvins, Barmarket, Soundgarden, ...
Hip-Hop	Neptunes, Obie Trice, Royce Da, Christina Milian, Alicia Keys, ...
Pop	Nelly Furtado, Madonna, Sugababes, Coldplay, Tori Amos, ...

Table 5. Genres and important artists (typical artists and genres of the used example collections)

units [4]. So there is good reason to expect quite good results[5] which, with a varying number of hidden units, are shown in Table (6). Note that the net

Model	Number of tracks
Linear Perceptron	34 per cent
MLP 1 Hidden Unit	ca. 22 per cent
2 Hidden Units	ca. 37 per cent
3 Hidden Units	ca. 46 per cent
4 Hidden Units	ca. 51 per cent
5 Hidden Units	ca. 57 per cent
6 Hidden Units	ca. 57 per cent
7 Hidden Units	ca. 59 per cent
8 Hidden Units	ca. 62 per cent (max)
9 Hidden Units	ca. 61 per cent
10 Hidden Units	ca. 61 per cent
11 Hidden Units	ca. 62 per cent
12 Hidden Units	ca. 60 per cent
16 Hidden Units	ca. 60 per cent

Table 6. Classification results for different numbers of hidden units

theoretically gets underdetermined when using five or more hidden units, so underdetermination is a far smaller problem than could have been assumed.

4.5 Comparison of the results:

#Genres	%Correct	#HU	Weights/Samples	Glm
4	75	3	3.22	81
9	62	8	5.30	34
21	40	20	2.34	20

Table 7. Comparison of classification results (number of genres, accuracy (mlp), number of hidden units, number of weights, accuracy of the linear perceptron)

Table (7) shows a summary of the results from the different training sets. I also calculated the ratio of the number of weights to the number of samples which turned out to be the most interesting part shown in the the last column named

[4] Following the simple rule that a network needs about ten times more training samples than its number of weigths. We have about 1600 samples and (9 plus 30) times *number of hidden units* weigths. 39 times 4 is 156, which times 10 is 1560, hence there's enough samples for this model up to about 4 hidden units (at least should be).

[5] 150 cycles, fifteen-fold cross-validation, varying number of hidden units

weights/samples. The higher this value the less under-determined the net (as mentioned before a net is under-determined below a ratio of about ten). Those values indicate that the downward slope of the classification results with an increasing number of genres is not (mainly) a problem of under-determination. My assumption is that the feature vectors are too similar. The results obtained with eight hidden units (so the maximum classification accuracy) are presented in greater detail. The confusion matrix computed for the validation data of the testcollection (4.4) shown in Table (8) verifies this assumptions. It also shows that the machine misclassification is quite similar to human misclassification (see how Reggae and Hip-Hop get misclassified for example). So accuracies between

x	Reg	Class	Metal	Ha	Slow.R.	Punk.R.	Grunge	Hip-Hop	Pop
Reggae	26	1	1	0	0	2	0	10	0
Class.	0	138	1	0	0	1	1	1	1
Metal	0	0	53	9	0	12	6	0	1
Hardc.	0	0	18	21	0	4	4	0	3
SlowR.	3	0	0	0	24	0	1	2	7
PunkR.	0	0	8	9	2	51	10	3	5
Grunge	3	4	6	3	5	3	37	3	5
Hip-Hop	29	8	2	0	2	1	1	144	22
Pop	0	3	1	7	14	4	9	15	48

Table 8. Confusion matrix (taken from validation set of testcollection (4.4), rounded average over different nets)

genres differ a lot, Table (9) shows the variance in accuracy across different genres. Actually there seems to be a correlation between classification accuracy

Genres	Class Accuracy
Reggae	0.6498
Classical	0.9665
Metal	0.6437
Hardcore	0.4163
Slow Rock	0.6487
Punk Rock	0.5715
Grunge	0.5293
Hip-Hop	0.6882
Pop	0.4703

Table 9. Classification results by genres

and the number of samples by class. This is shown in Figure (1). To prove this I tried with the same training set with equal size of the training data for the different classes (shrinked the size to 96 which is the smallest size of training

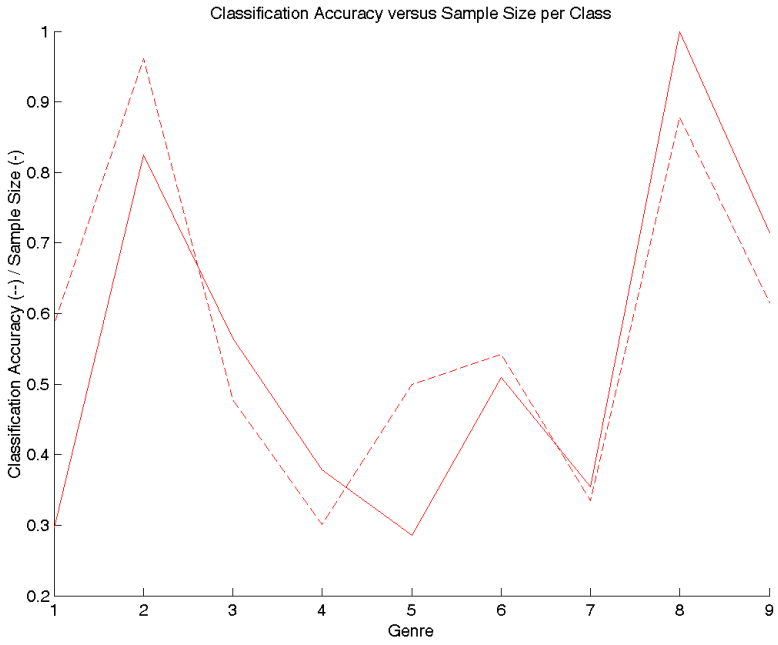

Fig. 1. Plot of the correlation between the percentage of sample size per class (-) and the classification accurracy per class (- -)

data for one class in this set). This collection consists of the same genres as the ones before but sample sizes are limited to 96 each, so the total sample size is 864. The model used to classify this collection is far more underdetermined than the one from section (4.4). Figure (2) shows a plot of the classification results by genres for both training sets. There is a very strong correlation between the two accuracies. So there is no significant correlation between different sample size and classification accuracy (the models just classify similarily). Figure (1) only shows that the collection by chance is composed of more samples of the genres that are best covered by the MARSYAS-features. Actually it only figures out that I had far more pieces of classical music than of the other ones and that classic is the genre that is best covered by the features (in other words: classical music is very different from anything else, at least in this collection). This leads to a short description of the role of the quality of the features. Figure

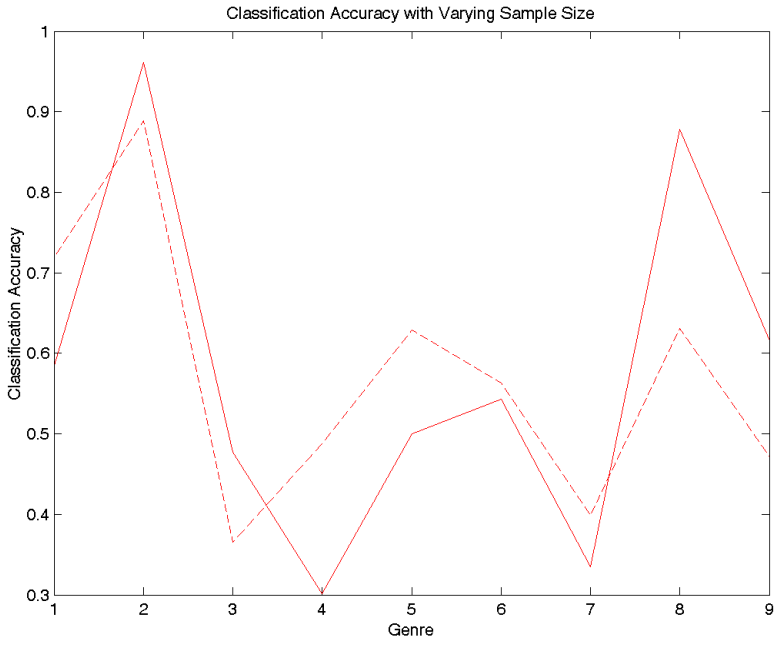

Fig. 2. Plot of the classification accuracy of a training set consisting of different sample sizes (-), compared to a set consisting of one sample size (96 samples for every class, - -)

(3) shows a plot of the mean of the features for Classical and Hip-Hop (the two genres with the highest classification accurary) whereas Figure (4) shows a plot of Pop and Hardcore features (the ones with the worst classification accuracy).

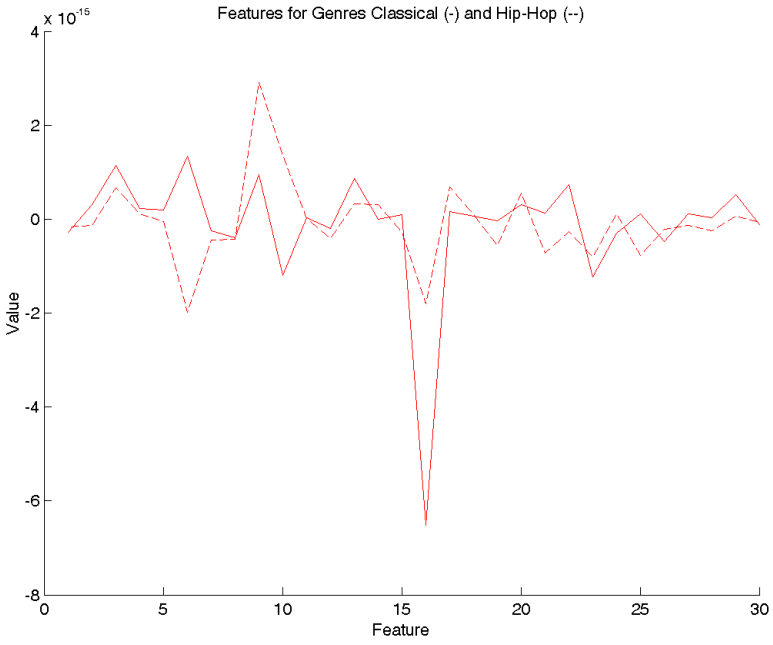

Fig. 3. Plot of the features for Classical (-) and Hip-Hop (- -)

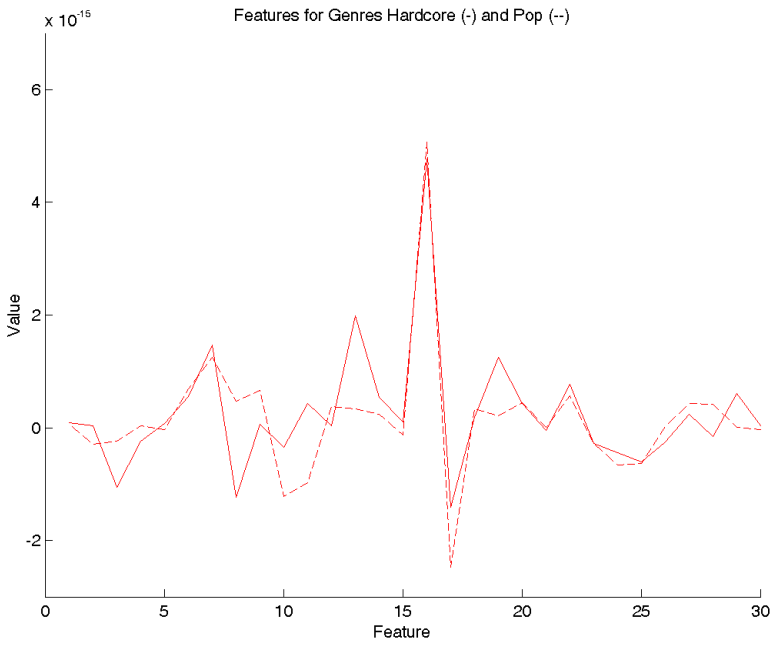

Fig. 4. Plot of the features for Pop (- -) and Hardcore (-)

Classical and Hip-Hop seem to differ more than Pop and Hardcore do, although I had hoped that the plots would point out that more clearly. The two lines do differ (in terms of the more different curves) more strongly for the plot for Classical and Hip-Hop in Figure (3) than in Figure (4) where Pop is compared to Hardcore.

5 Comparing to a Reference Set

Tzanetakis2002 [3] described quite similar experiments. Input data were computed by MARSYAS but classificatin of genres were done with the MARSYAS built in classification mechanisms: gaussian mixture models and k-nearest neighbour algorithm. His results were a classification accuracy of 61 per cent for 10 genres using a gaussian mixture model consisting of 5 gaussians. Quite comparable to the results of the previous section. To get a better comparison, I performed the same classification with Tzanetakis' reference set which consists of 1000 songs of 10 genres, hence a sample size of 100 each. Classification accuracy is 61.42 when using 9 hidden units and 15-fold cross-validation. So using a multi layer perceptron can compete with the gaussian mixture model.

6 Conclusions and Future Work

So what's wrong? The low (although it is not *that* bad) classification accuracy seems to be mainly a problem of features or feature quality. The article about wavelet packet representation ([5]) by Grimaldi et al. performs quite similar experiments using a different set of features. They achieve an accuracy of about 78 per cent (but only for five genres). The results are not easy to compare, but their feature set seems to be slightly better than Tzanetakis' because the experiments of Section (4) only classify about 72 per cent correctly for a set consisting of four different genres, whereas Grimaldi is around 78 per cent for five genres.

6.1 Samples

One could use samples of extended length (not from 30 seconds to sixty seconds of each audio file). Another interesting approach would be to take samples relative to the length of the audio file (e.g. starting at 35 percent, ending at 60 percent of an audio file). This would also imply a more fundamental inspecting of audio features. Those changes could possibly lead to better quality input data and therefore to better classifying results.

6.2 Features

Finding out the most important features and therefore decreasing the number of inputs would lower the samples to weight ratio and could lead to better

classification results. Classifying complete albums could be a nice application. Many albums contain tracks that are quite far from the overall genre of the album. A classification system could label a complete album according to the most chosen genre of its songs (an album consists of ten tracks, three classified as Rock, four classified as punk rock, three classified as Pop, the complete album is labelled as punk rock).

6.3 Limitations

Training sets do differ in size, a training set for one class is about 160 samples where it is only about 90 for another one (talking about Section (4.4)). Again: validation data is not completely independent of training data! Although this is more close to a real life scenario, it (probably) negatively affects the classification accuracy as seen in Figure (1). Just decided to not write another MatLab (and/or perl) script to split up into independent artist sets. Other interesting fields of work include the following:

- Getting better training data which shall include training sets with the same size for the different classes (e.g. 300 samples for each genre, the more, the better)
- Do a comprehensive analysis of the features)
  - Correlation analysis
  - Psychoacoustic exploration of important features
  - Combination with unsupervised learning
  - Working with genre-specific feature sets

Another interesting approach could be the hierarchical classification of music data. One could begin with separating any instrumental and vocal songs and then continue to differentiate between slow and more lively music. This could lead to a far better classification by the first layer (instrumental and vocal) and get poorer when going into greater detail.

References

1. AstonUniversity. Netlab, neural network toolbox for matlab. Internet, July 2004. www.ncrg.aston.ac.uk/netlab/.
2. George Tzanetakis, P. C. Marsyas: A framework for audio analysis. *Organised Sound 4(30* (2000).
3. George Tzanetakis, P. C. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* (2002).
4. Kosina, K. *Music Genre Recognition.* PhD thesis, Technical College Hagenberg, 2002.
5. Marco Grimaldi, Padraig Cunningham, A. K., Ed. *A Wavelet Packet Representation of Audio Signals for Music Genre Classification Using Different Ensemble and Feature Selection Techniques* (2003), MIR 2003.
6. Tao Li, Mitsunori Ogihara, Q. L., Ed. *A Comparative Study on Content-Based Music Genre Classification* (2003), SGIR 03.
7. Tzanetakis, G. Marsyas, a software framework for computer audition. Internet. http://marsyas.sourceforge.net/.