Automated musical instrument recognition

Alireza Amiri
Georg-August-University Göttingen
Institute of Computer Science
Göttingen, Germany
alireza.amiri@stud.uni-goettingen.de

Thorben Janz
Georg-August-University Göttingen
Institute of Computer Science
Göttingen, Germany
t.janz@stud.uni-goettingen.de

Dimitra Despoina Maoutsa
Georg-August-University Göttingen
Institute of Computer Science
Göttingen, Germany
d.maoutsa@stud.uni-goettingen.de

I. INTRODUCTION

The expanding volume of available musical data leads to a rapid growth of the research field of automated music information retrieval due to the need of efficient administration and exploration of musical content. This need results from the change of the way people listen to and interact with music. Modern music streaming applications enable the user to listen to music, which is selected based on tags the user indicates.

One important aspect of music information retrieval is the automated musical instrument recognition. It has several applications, such as automatic musical transcription. This process consumes a lot of effort and time for human beings. Therefore a tool supporting the extraction of musical scores is capable of saving time thus money.

The project discussed in this paper deals with the automated recognition of musical instruments. The goal is to hierarchically classify musical instruments according to their family (strings, woodwinds etc.) and furthermore, in case of success, to proceed to intra-family instrument distinction. This will be achieved by developing a classifier which uses the machine learning approaches. This project is part of the master's level class "Machine Learning and Pervasive Computing" at the Georg-August-University of Göttingen.

Obviously audio signals are needed as the input data. One alternative is recording audios by microphone. Another choice is using previously recorded audio files. In this project the second option is used due to the lack of high-quality devices. Therefore audio files represent the more suitable choice, to reduce misclassification caused by noisy data. Consequently no sensors are required to obtain the input data.

The majority of the research projects on this field is based either on the data of single note recordings or isolated solo-performances of single instruments, which are referred to as "monophonic" recordings. Only a small part of researches deals with polyphonic recordings, where two or more instruments are played simultaneously. In terms of recognizing instruments from monophonic recordings, most problems are solved accurately. Accordingly the present focus is set on polyphonic instrument recognition. In addition polyphonic recordings are closer to real-world music. For that reason polyphonic audio files are in this project's interest.

The novelty of our approach is the additional use of context information besides content information. Namely we are going to include information regarding the musical genre in order to provide some prior knowledge about the possible instrumentation of the songs of the recordings.

II. RELATED DOCUMENTS

The scientific work on automated musical instrument recognition has its roots in the late 70s. In this period of time, the basic methods and techniques for this classification problem were developed. The first attempts tried to map the sounds of musical instruments to a three-dimensional space, using subjective similarity judgments [1].

The research field had its height during the late 90s, when the existing methods were applied for processing manly monophonic audio signals. Based on the early works, the effort on recognizing instruments from single note recordings, respectively monophonic real-world recordings, was made [2]. Additionally, compared to the earlier attempts, the researches and projects used machine learning approaches to train their classifier.

Observations show that the accuracy of classifiers developed for monophonic signals, is significantly lower when applied on polyphonic recordings. In [3] three problems are stated, which need to be solved to achieve a high success rate in classification:

- feature variations
- timbre's dependency on the pitch
- musical context

The variation of features is the main problem when processing polyphonic audio signals. It is caused by the overlapping of and interference between the frequencies of the single instruments. Therefore it is not possible to separate the sounds of every single recorded instrument without distorting the acoustic signals. If this would be possible, the problem of polyphonic signal processing could be reduced to monophonic signal processing.

The approach to solve this problem which is mentioned in [3] gives every feature a weight. Features that are more likely to have high variance within a class due to overlapping frequencies get a lower weight. This is achieved by applying linear discriminant analysis.

The second issue, the timbre's dependency on the pitch, results from the wide pitch range of many instruments over many octaves. Therefore a pitch-dependent timbre model should be used to deal with this problem.

The consideration of neighboring notes when classifying the instrument playing a single note of a melody is a possible solution for the third problem. When identifying single notes of a melody, it can happen that the instrument playing one of these notes is classified differently as the others. Since this is very unlikely in musical way, a probability function should be taken into account, to decide if a likewise classification result is correct.

The paper [4] compares different techniques to separate different overlapping frequencies. This leads to the previous mentioned issue of feature variations. Consequently the more precise the technique separates the signals the lesser is this issue's impact. As it concludes, the FASST algorithm fits for separating the audio signals when attempting to recognize the predominant pitched instrument. It separates the input audio into the four parts:

- drums
- bass
- melody
- remaining sounds

The melody-part contains the signal of the predominant instrument separately. Accordingly this signal can be used as the input data for the classifier.

In [5] Mari Okamura proposed a novel approach called "Example-based Sparse Representation" in which source separation is not applied. This approach uses sample feature vectors of different musical instruments as the base matrix of sparse representation. The accuracy of 91.9% was obtained for monophonic sounds. Also in polyphonic sounds 51.1% of combinations were estimated correctly.

When attempting to differentiate between larger numbers of instruments, a used approach in research is the hierarchical classification approach. As described in [6] it uses several levels of abstraction, to particularize the classified instrument step by step. On one of the first levels, the decision may be based on the musical instrument family, while low-level decisions distinguish between the single instruments themselves.

Applying this approach, it is possible to construct subsets of features that are used on different nodes. Thus only well-suited features for the specialized decision in every node are considered by the classifier.

The crucial part applying this approach is the appropriate selecting of the features for deciding single steps. In addition, the way the different levels are built is important to achieve good results.

The usefulness of the hierarchical approach is proven by many projects in the past.

III. Data generation

As described in the introductory section, audio files of musical sound recordings are needed as the input data instead of self-recorded signals by means of a microphone.

Therefore the data generation limits itself on the retrieval of datasets, which are suitable and provided for research projects of that kind. Datasets that are taken into account must have several files with uniform attributes, both for the training and the test data. Since this project intends to recognize musical instruments of polyphonic recordings, the individual files must not contain the sound of only one single instrument. Additionally, the use of instrument simulating signals, e.g. MIDI files, should be avoided, to gain a mostly natural setting.

Ideally, the audio files are extracts of real-world songs. On the one hand, this guarantees a setting which is as natural as possible. On the other hand, the use of existing songs offers the possibility to use context information. Every accessible song can be related to one or several genres. By using this information, the set of instruments that probably can be contained in a certain audio file can be restricted to a specific subset. This is one crucial point of the project described in this paper.

Consequently, excerpts of songs from different genres are required. Otherwise the approach of using this information is not useful. Also, the considered genres ought to be selected from the "main" ones, because subgenres have mixed combinations of the main typical sets of instruments.

This leads to another requirement for the dataset utilized. There are two possibilities how to access information about one songs genre. First, the easier and more practical one, the data is annotated in advance with its genre. Possibly this information is integrated in the filename or the dataset contains a content descriptor. Alternatively, if the genre is not available, the extracts have to be appropriate to be recognized by a music identification service like Shazam or SoundHound.

IV. Data recorded and plan for features

The Music Technology Group of the University Pompeu Fabra of Barcelona provides datasets for musical instrument recognition on its website [7]. The datasets are separated into training and test data. The training data contains 6705 excerpts from about 2000 real-world songs. The instruments included in this dataset are the following:

- cello
- clarinet
- flute
- acoustic guitar
- electric guitar
- organ
- piano

- saxophone
- trumpet
- violin
- human voice

Since the audio files are extracted from real-world songs, the audio signals are polyphonic. Every excerpt has one of these instruments as the predominate instrument, which is annotated to the files by the filename. In addition, the filename of each file contains the information about the genre of the song. Following genres are mentioned:

- classic
- country-folk
- pop-rock
- jazz-blues

We omit the fifth genre "latin-soul", due to the lack of a sufficient number of excerpts for this genre.

Therefore, both the information about the instrument to be recognized and the associated genre are present for every audio file. The files are 16 bit stereo wav format files sampled at 44.1 kHz and are three seconds long.

The features planned to be extracted are listed below:

- autocorrelation coefficients: Evaluation of periodicities in signals
- zero crossing rates: number of times the signal changes sign
- spectral centroid
- spectral asymmetry/skewness: symmetry of the distribution
- spectral width
- spectral flatness: similarity of the power in all spectral bands
- MFCCs: representation of the short-term power spectrum of a sound
- RMS (root mean square): global energy of the signal
- crest factor: relation of peak values to the effective value
- spectral flux: measure of how fast the power spectrum of a signal is changing
- spectral roll off frequency: finding a frequency such that a certain fraction of the total energy is contained below that frequency
- spectral spread: standard deviation
- spectral slope: measure of how fast the spectrum of an audio sound tails off towards high frequencies
- spectral kurtosis: measure of the sharpness of the peaks

V. Feature Extraction

For the feature extraction procedure, we divide the raw musical signal in frames of length of 20 msec, with 50% overlap between consecutive frames. The temporal features (RMS,AC, ZCR) are calculated directly for each frame, while for the extraction of the spectral features, we perform a Discrete Fourier Transform for each frame and the features are extracted from the magnitude of the resulting spectrum. Consequently, for the extraction of the MFCCs from each frame, we apply a filterbank of 40 bandpass filters, equally spaced along the Mel frequency and then we get the log energy of each filter. Finally, we perform a Discrete Cosine Transform on the log energy obtained from the bandpass filters and calculate the resulting MFCCs. We decided to hold the 22 first coefficients in our feature vector.

For each of the aforementioned features, we calculate the mean, variance, maximum and minimum value over all the frames that consist one track.

For the feature selection part, we calculate initially the overall correlations between the extracted features in order to eliminate possible redundant dimensions. Consequently, we are going to use the wrapper model for feature selection, starting with the initial feature vector consisting only from the MFCCs – since according to the literature, these are the main descriptors of the timbre of a musical instrument.

VI. Classifier

As described in Chapter II we are going to apply the approach of hierarchical classification. That means that we divide the whole set of classes – musical instruments – into separate subsets. This division is mainly based on common musical instrument families.

- winds
- strings
- electrophones
- voice

The winds include the instruments clarinet, flute saxophone and trumpet. We do not subdivide the winds into woodwinds and brasses, because otherwise the subsets will not be big enough to benefit from the hierarchical approach.

The subset of strings contains the cello, acoustic guitar, piano and violin. Formally the electric guitar belongs to this family as well. However it is included by the third subset, the electrophones. This musical instrument family is intended for instruments, whose sound is created by the use of electricity. This third subset also contains one of the two remaining musical instruments, the organ. The organ in this case is only the electric version, not the pipe organ. Therefore the classification as an electrophone is justified.

The voice subset only contains the human voice, because it cannot be categorized in any other family according to the literature.

A second approach used for this project is the inclusion of the genre to improve the classification results. This is achieved by utilizing separated classifiers for each genre. That way each classifier can learn the sound of the musical instruments for its designated genre isolated from other genres. Thereby we try to decrease the frequency of misclassifications. Moreover the genre is taken into account in the classification process by giving probabilities of occurrence of the instruments within a certain genre.

After the genre detection the classifier distinguishes between the subsets of musical instruments as mentioned above for the specific genre. One step further, the intra-family instrument distinction is performed by another classifier. Summed up we make use of 16 single classifiers, one per genre for the subset detection and three per subset per genre for the specific instrument classification within a subset.

We use Support Vector Machines (SVM) for the classification, since they are widely used. Furthermore SVMs are useful for non-linear classification.

VII. Training and testing design

For gathering a separate dataset for training and testing, we divide the dataset into parts in ratio 4:1. This leads to datasets of 5366 tracks for training and 1339 tracks for testing the classifier.

VIII. Results achieved

The classification system is implemented using Matlab R2013a and additional toolboxes, named MIRToolbox 1.3.4 [8] and libsvm 3.2.0 [9]. These provide functionality for extracting features from musical audio signals respectively classification by Support Vector Machines.

The table given below illustrates the results achieved by the system.

	Cello	**Clari-net**	**Flute**	**A-Guitar**	**E-Guitar**	**Organ**	**Piano**	**Saxo-phone**	**Trum-pet**	**Violin**	**Voice**	∑	%
Cello	35	4	1	12	7	8	4	2	1	9	4	**87**	**40.23%**
Clarinet	0	20	9	6	4	12	14	26	10	5	4	**110**	**18.18%**
Flute	3	3	13	16	3	17	11	3	9	3	9	**90**	**14.44%**
Ac. Guitar	3	2	0	69	1	8	13	4	2	3	20	**125**	**55.20%**
El. Guitar	0	1	0	8	83	19	10	6	4	0	27	**158**	**52.53%**
Organ	0	2	1	5	12	88	2	3	3	0	21	**137**	**64.23%**
Piano	2	4	8	9	0	18	63	9	7	0	11	**131**	**48.09%**
Saxophone	1	8	1	10	12	15	11	27	12	4	10	**111**	**24.32%**
Trumpet	0	5	2	2	7	14	7	16	62	1	7	**123**	**50.41%**
Violin	8	4	2	6	14	12	4	10	8	40	16	**124**	**32.26%**
Voice	0	0	0	17	21	8	2	1	1	4	89	**143**	**62.24%**
∑	**52**	**53**	**37**	**160**	**164**	**219**	**141**	**107**	**119**	**69**	**218**	**1339**	**43.99%**

Table 1: Result of classification

Each row of Table 1 displays the classification results for one single instrument from the test dataset. The values in the diagonal are the number of test samples which are predicted correctly. In the last column the total accuracy of the classification of the single instruments are listed. The green cell represents the total accuracy of the classification system.

The overall accuracy amounts 43.99%, while the best results are achieved for the organ and the voice. The worst accuracy is gained for the flute and clarinet.

What is noticeable is the proportionally high precision of the classification of instruments belonging to the electrics family. In the contrary, the accuracy of the prediction for the instruments of the winds family is relatively lower. This leads to the assumption that a high amount of samples that actually belong to the winds family are falsely classified as another family in the preceding family distinction.

To analyze this assumption the following table gives the results of the family prediction.

	strings	**electrics**	**winds**	**accuracy**
strings	280	119	68	59,96%
electrics	48	368	22	84,02%
winds	94	114	226	52,07%

Table 2: Result of family classification

As Table 2 shows the family classification of the electric family is relatively accurate compared to both other families.

To exclude that the low accuracies result from single instruments, the table below gives the result of the family classification for every single instrument.

	strings	electrics	winds	right	wrong	accuracy
Cello	60	19	8	60	27	**68,97%**
Clarinet	25	20	65	65	45	**59,09%**
Flute	33	29	28	28	62	**31,11%**
A-guitar	88	29	8	88	37	**70,40%**
E-guitar	18	129	11	129	29	**81,65%**
Organ	7	121	9	121	16	**88,32%**
Piano	74	29	28	74	57	**56,49%**
Saxophone	26	37	48	48	63	**43,24%**
Trumpet	10	28	85	85	38	**69,11%**
Violin	58	42	24	58	66	**46,77%**
Voice	23	118	2	118	25	**82,52%**

Table 3: Result of family classification per instrument

Since no instrument differs extremely from the average accuracy of its family, the overall performance of the family prediction causes the results stated in Table 2.

Therefore the focus for improving the results of the classification system described is set on the family classification process, especially concerning the strings and winds family.

IX. IMPROVED RESULTS

As mentioned in the previous section, the focus for improving the results is set on the family classification process. Small improvements were achieved by selecting better performing feature sets. However in parallel the results of the intra-family classification became worse.

Therefore different other approaches were applied for the improvement. First, all the feature were normalized to the range of 0-1. This influences the classifier in the way that within a family most test instances got classified as one single instrument. Consequently the performance got worse.

This behavior appears to be caused by an unbalanced training dataset. To compensate this problem, we calculated several parameters passed to the SVM during the training phase, as class-weights and cost values that penalize the points that lay between the hyperplanes. These values are calculated by running the training and testing process several times with different values within a defined range and selecting the values from the best performing configuration. Surprisingly, using these parameters deteriorate the results as well.

Under these circumstances we cannot improve the overall result in an appropriate amount of time.

X. REFERENCES

[1] J. M. Grey, „Multidimensional perceptual scaling of musical timbres,“ Center for Computer Research in Music and Acoustics, Department of Music, Stanford University, Stanford, 1976.

[2] G. De Poli, P. Prandoni und P. Tonella, „Timbre clustering by self-organizing neural networks,“ University of Milan, Milan, 1993.

[3] T. Kitahara, M. Goto, K. Komatani, T. Ogata und H. G. Okuno, „INSTRUMENT IDENTIFICATION IN POLYPHONIC MUSIC: FEATURE WEIGHTING WITH MIXED SOUNDS, PITCH-DEPENDENT TIMBRE MODELING, AND USE OF MUSICAL CONTEXT,“ Dept. of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Kyoto, 2005.

[4] J. Bosch, J. Janer, F. Fuhrmann und P. Herrera, „A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals,“ Music Technology Group, Universitat Pompeu Fabra, Barcelona, 2012.

[5] M. Okamura, M. Takehara, S. Tamura und S. Hayamizu, „Toward polyphonic musical instrument identification using example-based sparse representation,“ Department of Information Science, Gifu University, Gifu, 2012.

[6] A. Eronen und A. Klapuri, „Musical instrument recognition using cepstral coefficients and temporal features,“ Signal Processing Laboratory, Tampere University of Technology, Tampere, 2000.

[7] Music Technology Group - Universitat Pompeu Fabra - Barcelona, „IRMAS: A DATASET FOR INSTRUMENT RECOGNITION IN MUSICAL AUDIO SIGNALS,“ [Online]. Available: http://mtg.upf.edu/download/datasets/irmas. [Zugriff am 2 May 2015].

[8] University of Jyväskylä, „MIRtoolbox — Humanistinen tiedekunta,“ 2014. [Online]. Available: https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox. [Zugriff am 22 06 2015].

[9] C.-C. Chang und C.-J. Lin, „LIBSVM -- A Library for Support Vector Machines,“ 2014. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/libsvm/. [Zugriff am 22 06 2015].