Musical Instrument Detection

Detecting instrumentation in polyphonic musical signals on a frame-by-frame basis

Greg Sell Gautham J. Mysore Song Hui Chon

Center for Computer Research in Music and Acoustics

December 15, 2006

1 Overview

Knowledge of the instrumentation of a musical signal at any given time could be useful for major audio signal processing problems such as sound source separation and automated music transcription. Knowing which instruments are playing is a first step toward more intelligently designed solutions to these very important and largely unsolved challenges.

So, in this paper, we attempt the problem of identifying the instrumentation of a musical signal at any given time using several machine learning techniques (logistic regression, K-NN, SVM). We approached the problem as a series of separate binary classifications (as opposed to a multivariate problem) so that we could mix and match the best algorithm for each instrument to create the best overall classifier.

2 Data

Data collection and labeling in audio research can be extremely problematic. Hand labeling audio frames (of lengths of 20 ms) is very difficult and time consuming. So, in order to expediate this process, we chose to artificially manufacture our polyphonic music by combining solo recordings of each instrument.

We chose to investigate 3 combinations of instruments:

- Piano, Violin
- Clarinet, Piano, Violin
- Cello, Clarinet, Flute, Piano, Violin

We collected 4 samples each from 13 different recordings for each of the five instruments (leading to 52 total signals for each instrument). The samples were each between 1 and 6 seconds in length, and consisted only of the specified instrument playing in solo (no silence). Then, to create one of the combinations above, a random signal for each instrument were all combined randomly. In this way, we created 52 total signals for each instrumental combination. The final 12 signals of these 52 were created from recordings completely independent of those for the first 40, giving our training (the first 40) and test (the final 12) data.

We used examples from multiple recordings in order to create a more robust system. An instrument will have many unique features on a given recording. Every individual instrument has a unique character (due to materials. construction process, body shape, age, etc.), and every individual performer creates a different sound with the instrument. There are also many different playing techniques, and then microphone choice and positioning as well as digital effects and equalization will all create significant variations between recordings. So, in order to effectively analyze an arbitrary recording, we need to train our classifiers with multiple examples.

The test data was created from recordings independent of the training data in order to simulate the real-life situation of encountering a recording with new filtering, players, etc.

3 Features

The data was first segregated into frames of 1024 samples each (23 ms). This frame size was selected for its wide use in speech processing applications.

Once divided, we chose to describe each frame with three types of features, which were decided based on acoustic knowledge of the instruments:

- Magnitudes of the *Discrete Fourier Transform* (DFT)
- *Mel Frequency Cepstral Coefficients* (MFCCs)
- Change in energy from frame to frame

3.1 DFT Magnitude

Every instrument has defining characteristics in the frequency domain. Frequency range and frequency bandwidth are often indicative of an instrument. Characteristic frequency peaks independent of a note's pitch, known as formants, are also often present in the spectrum of a certain instrument.

In hopes of exploiting these types of differences, we included the magnitude coefficients of the DFT of each frame. In order to reduce the size of the data set, we averaged the coefficients in 4 neighboring bins to reduce the resolution. We also cut off the analysis in the 84^{th} bin, or 3620 Hz, because visual analysis of spectrograms determined there was not enough energy present above that point to warrant inclusion of the data.

3.2 MFCCs

MFCCs are used extensively in speech and speaker recognition. Essentially, they represent the *Discrete Cosine Transform* of the log spectrum of a signal analyzed on an auditory frequency scale (the Mel scale). The process creates a 13-dimensional vector that summarizes the signal's spectrum.

We included MFCCs to represent the differences in the shape of the spectrum for different signals (which cause the timbral differences that provide one easy means for the ear to differentiate the signals).

3.3 Energy Change

Different instruments also behave differently over time. Some instruments attack quickly and then decay exponentially from there. Other instruments attack slowly then decay quickly. Every instrument has a unique amplitude envelope over time (known in synthesis as the *ADSR*, or Attack, Decay, Sustain, Relief).

The energy changes of the previous 6 frames were included as a result of this. This feature set will indicate the most recent tendencies in the energy change, showing whether there has recently been a sudden attack, slow attack, sudden decay, or slow decay (or some combination).

4 Algorithms

Using these features, or some subset (see below), we attempted the problem with four different methods:

- Logistic Regression
- K-Nearest Neighbors (K-NN)
- Support Vector Machine (SVM) with Linear Kernel
- SVM with Gaussian Kernel

4.1 Logistic regression

Logistic regression was implemented using standard gradient descent. The data was cycled through multiple times using a decaying α parameter (decaying from .1 to .0001) to provide a quick convergence.

4.2 K-NN

K-NN was implemented with the SOM toolbox [1]. For every instrument in every combination, every value of K from 1 to 2000 was considered. Selection of the K parameter is described in the Training and Testing section.

Distances between points were calculating using the euclidean norm. In order to minimize the domination of a given feature, we preprocessed the data to unit variance in all features.

4.3 SVM

SVMs with each kernel were implemented using Spider [2]. For each instrument, a model selection was performed separately with each of the kernels.

For the linear kernel, 10 logarithmically spaced values of C were used. For the Gaussian kernel, 10 logarithmically spaced values of C and 10 logarithmically spaced values of σ (a total of 100 combinations of parameters) were used.

A larger number of parameter values would have been desirable but due to high computation time (some of the matlab scripts took well over 24 hours to run), we decided to stick with the current set of parameter values. Logarithmic spacing of parameter values has been used so that a larger range of values could be spanned.

Selection methods are described in the Training and Testing section.

5 Training and Testing

As described in the Data section, the data was split into two completely independent pools, designated training and testing. However, due to the high computation cost of training SVMs, we were forced to reduce the data size and feature set to yield reasonable computation times (around 24 hours). So, in the case of SVMs, only the MFCCs and temporal energy changes were used (DFT coefficients were excluded). Also, in each case, a trimmed data set was used for training. In the case of 2 instruments, signals 1 to 30 were used. For 3 instruments, signals 1 to 25 were used. And, for 5 instruments, 1 to 15 were used.

In all cases, the testing set was identical.

For logistic regression and K-NN, the algorithms were run in two different situations. In one case, the algorithms were trained and tested on the same limited data set and feature set as the SVMs. From here, those will be referred to as limited logistic regression and limited K-NN.

The algorithms were then run again using all of the available data (full training set, full features). Those cases will be referred to as full logistic regression and full K-NN.

5.1 Model Selection - SVM and K-NN

In the case of SVM and K-NN, it was necessary to perform model selection to find the values for C, K, and σ. For SVM and limited K-NN, we used 70-30 cross validation to select the model, dividing the limited training set into the selection training and selection testing sets. The parameters were then selected as the values that achieved the best error rates.

For full K-NN, 70-30 cross validation was again performed, but this time using the full training and feature sets. These selected parameters are displayed in Fig. (1) for K-NN and Fig. (2) for SVM.

It should also be noted that, as stated before, the parameters for the SVMs were not able to be optimized beyond one pass over a wide logarithmic scale.

5.2 Results

Once model selection was complete, we predicted the instrumentation in the test signals. In the case of logistic regression and K-NN, separate predictions were made using the limited and full cases. Tables containing test accuracy for each instrument can be found in Fig. (3) for limited cases and Fig. (4) for full cases.

	Limited K-NN	Full K-NN
Piano	28	36
Violin	19	513
Clarinet	309	78
Piano	18	74
Violin	1883	48
Cello	1817	9
Clarinet	1966	177
Flute	35	268
Piano	260	112
Violin	1691	1508

Figure 1: K values selected for K-NN

The differing computational costs of the algorithms prevent us from simply determining the best overall approach for each case, because there are two ways to view the data: one is to consider all the algorithms on the limited scale to assess the best performance on the same data, and the other is to compare the SVMs to the full logistic and K-NN to determine the best option with a realistic computation time. We analyzed the data in both ways.

5.3 Limited Case Analysis

SVMs, in general, give strong results with both kernels. The linear kernel is at or near the top performance for every classification except clarinet, cello, and piano in the five instrument case. The gaussian kernel is at or near the top performance for every classification except clarinet in the three instrument case, and flute, clarinet, and piano in the five instrument case.

	SVM-Linear	SVM-Gaussian	
	C	C	σ
Piano	10000	59.9484	12.9155
Violin	0.006	1	12.9155
Clarinet	0.001	0.001	0.01
Piano	215.4435	1000	35.9381
Violin	0.001	1000	12.9155
Cello	0.01	21.5443	12.9155
Clarinet	0	0.0001	0.01
Flute	1	21.5443	12.9155
Piano	0	0.2154	12.9155
Violin	0.0005	0.0464	1.6681

Figure 2: Parameters selected for SVM

	Logistic	K-NN	SVM-Lin.	SVM-Gaus.
Pf	0.7742	0.8991	0.8947	0.8954
Vl	0.8538	0.6923	0.8755	0.8723
Cl	0.6899	0.7018	0.7327	0.6581
Pf	0.7295	0.7122	0.7835	0.7889
Vl	0.7232	0.7816	0.7580	0.7669
Vc	0.6678	0.7006	0.6885	0.7368
Cl	0.7515	0.7137	0.7137	0.7137
Fl	0.7985	0.8037	0.8229	0.5869
Pf	0.6958	0.6564	0.6558	0.6494
Vl	0.7541	0.7791	0.7961	0.8049

Figure 3: Test accuracy of limited algorithms. (Vc - Cello, Cl - Clarinet, Fl - Flute, Pf - Piano, Vl - Violin)

Limited logistic regression comfortably outperforms all other systems for the clarinet and piano in the five instrument case.

The only cases where limited K-NN outperforms the other algorithms are piano for two instruments and violin for three instruments. However, in both cases, SVMs are able to get similarly accurate results.

5.4 Full Case Analysis

In the two instrument case, the full logistic regression and full K-NN are outperformed by the limited algorithms. However, as the instrument combinations become more complex, the full algorithms become more and more effective. The most significant improvements are with the clarinet and piano in the three instrument case and with the piano in the five instrument case, where the full algorithms greatly outperform the limited algorithms.

	Logistic	K-NN
Pf	0.8369	0.8732
Vl	0.8686	0.8665
Cl	0.7333	0.7705
Pf	0.8126	0.7825
Vl	0.7529	0.7859
Vc	0.7499	0.7433
Cl	0.7489	0.7523
Fl	0.7886	0.8318
Pf	0.7459	0.7593
Vl	0.8078	0.8171

Figure 4: Test accuracy of full algorithms. (Vc - Cello, Cl - Clarinet, Fl - Flute, Pf - Piano, Vl - Violin)

Even though full K-NN is often the best classifier, it is an extremely data inefficient algorithm. In order to make a prediction, the new frame must be compared to every training frame and then sorted. The other algorithms are much more efficient in this regard, so in applications were data storage or computation speed are important, K-NN should not be considered.

6 Conclusions

Fig (5) shows the test accuracy attained by combining the best classifiers, both in the case of only considering limited algorithms and in the case of considering all algorithms. We were able to make this step because of our initial decision to treat the problem as separate binary classifiers.

In the two instrument case, we are able to predict both instruments with an accuracy of 79.46%. In the three instrument case, we were able to predict all three instruments 51.69% of the time (with 8 possible combinations). In the five instrument case, we predicted all five instruments correctly 42.44% of the time (with 32 possible combinations).

The table also shows how often the classifiers correctly predict at least a certain portion of the combination correctly. In some applications, this less restricted level of accuracy could be as important.

6.1 Most Common Mistakes

Fig. (6) shows the instrument combinations that were most common in false positive mistakes for each classifier. Fig. (7) shows the combinations the were most common for false negatives.

6.2 Future Work

For our future work, we would like to find either more efficient SVM software or more powerful computers so that we could experiment with feature selection, which will be especially important as we attempt to augment and refine our feature set. Ideally, we could even select different feature sets for each instrument's classifier, as was seen in [3] (though, in this case, each possible combination of instruments was given its own binary classifier, each of which using an individually optimized feature set).

Also, we would like to correct two trends that we see in our results. One is that our overall accuracy

# Correct	Limited Best	Overall Best
1	0.9801	0.9801
2	0.7946	0.7946
1	0.9843	0.9927
2	0.8518	0.8594
3	0.4671	0.5169
1	0.9948	0.9983
2	0.9564	0.9718
3	0.8337	0.8618
4	0.6316	0.6543
5	0.3955	0.4244

Figure 5: Test accuracy for Limited Best and Overall Best for different number of instruments required to be correct.

drops a great deal in moving from two instruments to three, but then remains about the same in the move from three to five instruments. The other is that there is a great deal of consistency in the most commonly mistaken combinations across the different algorithms.

Both of these trends indicate to us that we need to expand/improve our feature set. The fact that the classifiers all make similar mistakes suggests that the classes are not properly resolved in the feature space in those cases. Also, the similarity in performance

Obviously, an expansion of the training data would also help. Some of the less favorable patterns in our results could most likely be improved by exposing the classifiers to more instances of those particular combinations, and adding more training data should improve the performance. The vast improvement in logistic regression and K-NN from the limited case to full case demonstrates that increasing the data size and feature set can both yield significant results.

	L-Log.	L-K-NN	SVM-L	SVM-G	F-Log.	F-K-NN
Pf	Vl	Vl	Vl	Vl	Vl	Vl
Vl	Pf	Pf	Pf	Pf	Pf	Pf
Cl	Pf,Vl	Pf,Vl	Pf,Vl	Pf	Pf,Vl	Pf,Vl
Pf	Cl	Cl	Cl	Cl	Cl	Cl
Vl	Cl	Cl,Pf	Cl,Pf	Cl,Pf	Cl,Pf	Cl,Pf
Vc	Pf	Pf	Pf	Pf	Pf	Pf
Cl	Pf	-	-	-	Fl	Fl
Fl	Vc,Cl	Vc,Cl	Vc,Cl	Pf	Vc,Cl	Vc,Cl
Pf	Fl	-	-	Vc,Cl	Vc,Cl	Fl
Vl	Cl	Vc,Cl	Vc,Cl	-	Vc,Cl	Vc,Cl

Figure 6: Most commonly mistaken combinations for false positives. (Vc - Cello, Cl - Clarinet, Fl - Flute, Pf - Piano, Vl - Violin)

	L-Log.	L-K-NN	SVM-L	SVM-G	F-Log.	F-K-NN
Pf	Pf,Vl	Pf	Pf,Vl	Pf,Vl	Pf,Vl	Pf,Vl
Vl	Pf,Vl	Pf,Vl	Pf,Vl	Pf,Vl	Pf,Vl	Pf,Vl
Cl	Cl	Cl	Cl	-	Cl	Cl
Pf	Pf,Vl	Cl,Pf,Vl	Pf,Vl	Cl,Pf,Vl	Cl,Pf	Pf,Vl
Vl	Cl,Pf,Vl	Cl,Pf,Vl	Pf,Vl	Pf,Vl	Pf,Vl	Pf,Vl
Vc	Vc	Vc	Vc	Vc,Pf	Vc,Pf	Vc,Pf
Cl	Cl	Cl	Cl	Cl	Vc,Cl	Cl
Fl	Fl	Fl	Fl	Fl	Fl	Fl
Pf	Pf	Pf	Pf	Pf	Pf	Pf
Vl	Vc,Vl	Vc,Vl	Vl	Vc,Vl	Vc,Vl	Vc,Vl

Figure 7: Most commonly mistaken combinations for false negatives. (Vc - Cello, Cl - Clarinet, Fl - Flute, Pf - Piano, Vl - Violin)

However, in order to most efficiently enhance the performance of our classifiers, refining the feature set is most necessary.

References

[1] Juha Parhankangas Esa Alhoniemi, Johan Himberg and Juha Vesanto. *SOM Toolbox.* availabe online at http://www.cis.hut.fi/projects/somtoolbox, 2000-5.

[2] Gokhan Bakir Jason Weston, Andre Elisseeff and Fabian Sinz. *The Spider.* available online at http://www.kyb.tuebingen.mpg.de/bs/people/spider/, 2006.

[3] G. Richard S. Essid and B. David. Instrument recognition in polyphonic music. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 2005.