Clustering Spectral Filters for Extensible Feature Extraction in Musical Instrument Classification

Patrick J. Donnelly and **John W. Sheppard**
Department of Computer Science
357 EPS Building
Montana State University
Bozeman, MT 59717
{patrick.donnelly2, john.sheppard}@cs.montana.edu

Abstract

We propose a technique of training models for feature extraction using prior expectation of regions of importance in an instrument's timbre. Over a dataset of training examples, we extract significant spectral peaks, calculate their ratio to fundamental frequency, and use k-means clustering to identify a set of windows of spectral prominence for each instrument. These windows are used to extract amplitude values from training data to use as features in classification tasks. We test this approach on two databases of 17 instruments, cross evaluate between datasets, and compare with MFCC features.

Introduction

Musical instrument classification is an important task necessary to music search, genre identification, and automatic score transcription. While there have been many approaches to recognize individual instruments, the majority of these are not extensible to the more complex case of identifying instruments present in polyphonic mixtures.

This paper presents an approach for feature extraction that is designed to be scalable to the task of instrument recognition within polyphonic mixtures. Over a large dataset of instruments, we extract significant peaks and cluster the ratios of these peaks into a spectral signature that informs which harmonic locations contain spectral energy. We use these locations as spectral filters to perform feature extraction for a classification task. We evaluate this feature extraction scheme on two datasets using four classifiers. We compare our feature extraction scheme to the commonly used set of Mel Frequency Cepstral Coefficients (MFCC) features.

Related Work

Although many studies have explored the recognition of musical instruments playing isolated notes, no dominant learning strategy nor feature extraction technique has emerged.

A variety of supervised classification techniques have been explored, including k-nearest neighbors, support vector machines, decision trees, Gaussian mixture models, Bayesian networks, linear discriminant analysis, and neural networks (see Herrera-Boyer, Peeters, and Dubnov 2003 for review). Additional studies have explored various spectral, temporal, and cepstral features for instrument recognition (see Deng, Simmermacher, and Cranefield 2008 for review).

Many of the techniques attempted in solo instrument classification are not practical for classification of real music performances in which multiple instruments often play at the same time. The task of recognizing instruments present in polyphonic mixtures is a more complex task as the harmonics of the instruments are interleaved in both time and frequency. Unfortunately, many of the feature extraction approaches attempted for single instrument classification are not extensible to the polyphonic mixture task.

Essid, Richard, and David (2006) created a system that does not require source separation but uses hierarchical clustering to build a taxonomy of instruments playing simultaneously, achieving 53% accuracy on a dataset of jazz recordings. These experiments train on fixed combinations and are not extensible to unseen combinations of instruments.

Most approaches to instrument recognition in polyphonic music attempt a form of source separation. In an attempt to minimize source interference, Kitahara et al. (2007) used linear discriminant analysis to minimize the weight of features most affected by overlapping partials in polyphonic mixtures of sounds. On a dataset of mixtures of five instruments, the authors achieved 84% accuracy for duets, 77% for trios, and 72% for quartets. Leveau et al. (2008) decomposed signals into a mid-level representation to train a dictionary of prototypical atoms based on solo instrument examples. The authors model signals as the composition of various pitch and instrument specific atoms using an optimization process, achieving between 56% and 87% accuracy in a single instrument recognition task over a dataset of five instruments.

Another approach, inspired by computational auditory scene analysis, uses sinusoidal modeling and dimensionality reduction to build prototypical spectro-temporal envelopes of different instruments. One study used a graph partitioning algorithm to cluster these envelopes and classify a set of six instruments, ranging from 83% accuracy in the single instrument case to 56% for four instrument mixtures (Martins et al. 2007). Another study modeled these envelopes as Gaussian processes and used Euclidean distance to the prototypes as a classification metric, achieving 94.9% accuracy for single instruments and 54% for four instrument mixtures on a set of five instruments (Burred, Robel, and Sikora 2010).

Design Goals

Many of the classification approaches and feature extraction techniques for instrument recognition cannot be extended to the recognition of instruments in polyphonic mixtures. We identify the following design criteria needed to extend an approach to multi-label classification of polyphonic mixtures. In this work, we design a feature extraction scheme with extensibility to multilabel classification and we evaluate our scheme on the single instrument classification task.

Scalability

The goal of the identifying instruments present in polyphonic mixtures is a multi-label classification problem. One approach is to train on all possible mixtures of instruments as single classes, taken by Essid, Richard, and David (2006). This method, however, suffers from the combinatorial explosion of labels needed to classify and it is not feasible to train models with every possible combination of instruments.

The task of polyphonic identification lends itself naturally to binary relevance (BR) classification, a decomposition approach in which a single classifier is trained for each instrument in order to identify the presence of that instrument in a signal, independent of any other instruments that may be present (Luaces et al. 2012). The strength of BR classification for polyphonic mixture identification is that it only requires training models on single instrument data yet allows extensibility to unseen combinations of those instruments.

In this work, we cluster models for each instrument that inform the locations in the harmonic spectra most often containing significant spectral energy. We use these instrument-specific signatures to filter strategic windows in each example's spectra and extract amplitude values as features. For each instrument we then train a binary classifier.

Generalizability

Arguing that many approaches cannot generalize to new data, Livshin and Rodet (2003) identified five different musical instrument datasets that shared a common subset of seven instruments and performed cross database evaluations. The authors received results ranging from 20% accuracy in the worst case up to 63% in the best, with an average accuracy of 42%, demonstrating the poor generalization abilities of common classification techniques across databases.

In this work, we compare cross dataset performance on two common datasets. These datasets feature multiple performers, instrument manufacturers, articulations, dynamic levels, and cover the range of each musical instrument.

Practicality

Timbre perception and recognition relies on both the harmonic content of the musical partials and the fine timing of the envelope of each harmonic. The attack of an instrument sound and the differences in the fine-timing of the envelopes of individual partials are of particular importance in both perception and algorithmic recognition of timbre. Many classification approaches exploit this valuable information, as does the human auditory system (Fuhrmann, Haro, and Herrera 2009).

The literature has focused on single instrument classification in which the datasets contain examples of the entire length on an instrument sample, including the attack and the decay. An approach that relies on the time differences of the instrument's envelopes may not scale well to situations in which signals contain only part of an instrument's note.

In this work, we will ignore any timing information and instead focus on identifying locations of harmonic content most useful in discriminating between musical instruments.

Feature Extraction

Our feature extraction scheme relies on spectral features in order to estimate source separation. Except in cases of interfere caused by overlapping partials, musical partials are generally well separated in the frequency domain (Figure 1).

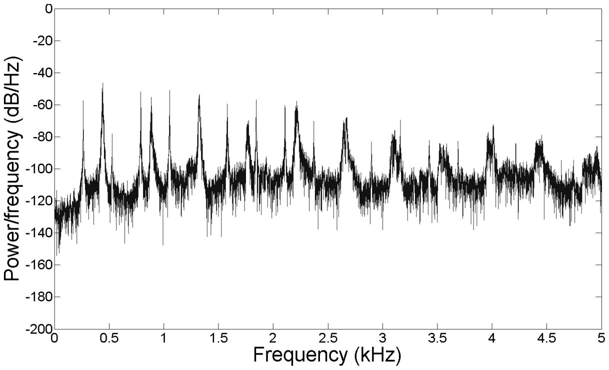

Figure 1: Overlapping spectra of a Clarinet (261 Hz) and a Violin (440 Hz).

Our approach discovers locations of harmonic content, normalized by the fundamental frequency, that are dominant across a large set of examples for each instrument. The locations are used as spectral filters to extract amplitude values in the location strategic to the instrument, and these values are used as features in classification.

Signal Processing

First, a high pass filter with a cutoff of 20 Hz is applied to each file to eliminate low-frequency noise. Next, a Fast Fourier Transform (FFT) with a single time window the entire length of the recording transforms the waveforms to the frequency domain. The resulting spectral magnitudes are scaled by $10 \cdot \log 10$ dB to convert from a linear scale to a Power/Frequency scale.

Spectral Threshold

Spectral peak detection is necessary to detect and extract the harmonics in the spectra. For each example, we define a frequency-dependent threshold and identify all spectral peaks that exceed it. We use a variable frequency-dependent threshold to capture the amplitudes of higher harmonics, despite the roll-off found at higher frequencies.

We employ the thresholding strategy presented by Every and Szymanski (2006). First, a smoothed amplitude envelope E is calculated for each example by convolving the spectra F with a moving Hamming window h of length $256 + 1$ samples in which each value of E_j is set to be the weighted average of the window with center point j. We choose an odd-length window for symmetry.

The frequency-dependent threshold for each frequency bin j is calculated as

$$\hat{E}_j = e_{th} \cdot (E_j)^c \tag{1}$$

where c is a constant $[0.5, 1)$ that determines the flatness of the envelope shape and e_{th} is a frequency independent threshold height. The parameter e_{th} is defined as

$$e_{th} = b \cdot \mid \overline{F} \mid^{1-c} \tag{2}$$

where $\overline{F}$ is the average amplitude across all frequency bins and b is a positive scalar that raises the mean above the noise floor. We choose $c = 0.5$ to produce a flatter envelope and a value of $b = 4$ in all our experiments. An example spectrum with threshold $\hat{E}$ is shown in Figure 2.

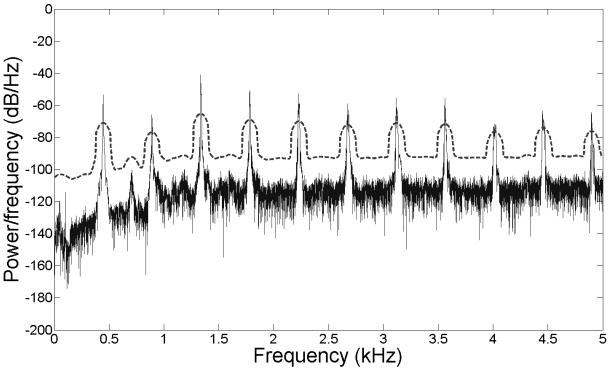

Figure 2: Amplitude spectrum (solid) overlaid with threshold (dotted) of a Clarinet playing A4 (440 Hz). For readability, the depicted spectrum is limited to the first 5 kHz.

Spectral Peak Identification

For each example, we extract peaks in all spectral bins that exceed the threshold, $F_j > \hat{E}_j$, for all frequency bins j up to the Nyquist limit, and save the corresponding frequency in the vector $\mathbf{p}$. The amplitude value in bin F_j is discarded because we are interested in identifying the ratios of these spectral peak locations to the fundamental frequency and we must detect the fundamental frequency f_0 for each audio file. Since we are training on files containing only single instruments, we employ a näive f_0 finding algorithm in which a frequency bin j is considered to correspond to f_0 if

$$\underset{j}{\operatorname{argmin}}\{\forall k \in (1,32): \; F_{j-k} < F_j > F_{j+k} \wedge F_j > \hat{E}(j)\} \tag{3}$$

In other words, f_0 corresponds to the frequency value for the lowest frequency bin j that contains the highest amplitude value F_j within a localized window of 32 samples and also exceeds the threshold $\hat{E}(j)$. For each peak $p \in \mathbf{p}$, its ratio to f_0 is calculated as $r = p/f_0$. Any ratio $r > 64$ is discarded and the rest are saved in a vector of ratios $\mathbf{r}$.

Signature Clustering

k-means is a common clustering algorithm that partitions a set of n observations into k discrete clusters so that every observation is assigned to the cluster with the nearest mean (Bottou and Bengio 1995). We use k-means to inform the locations of Gaussian clusters at various harmonics for each instrument.

Over the two datasets and for each instrument, the ratios $\mathbf{r}$ are extracted. These vectors are concatenated into a single one-dimensional vector, with duplications permitted. This vector is passed to a k-means clusterer. We used a fixed number of clusters k, which we vary experimentally, and ran k-means until convergence. To reduce convergence time of the algorithm, we seed the initial k clusters with values $[2 \ldots k + 1]$ to correspond to our expectation that most clusters will contain means near integer ratios of the fundamental.

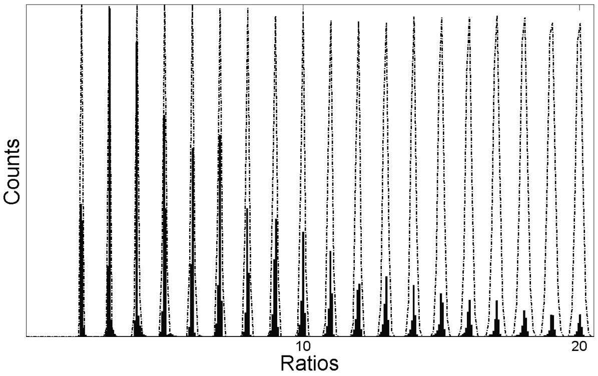

Figure 3: Histogram of spectral peaks (solid) overlayed with clusters learned (dotted) for the set of Clarinet examples in the RWC dataset. For clarity, only the clusters with ratios ranging from 2 up to 20 are shown.

For each cluster, we extract the mean and standard deviation. This set of cluster means and standard deviations are saved for each instrument and will be used to determine the location of the harmonics for feature extraction on an instrument-specific basis. Cluster density information, illustrated in Figure 3 as a histogram around ratios, is not used in these experiments. This valuable information will be used in future work to prune clusters.

Feature Extraction

The set of clusters learned from k-means is used as a spectral filter to decide the specific locations from which to extract amplitude information. We extract a single amplitude value for each cluster. For each example, f_0 is identified using the method described above. Given a cluster c_i with mean c_i^μ and standard deviation c_i^σ, we identify a window w_i corresponding to bins containing the frequencies in the range $[((c_\mu^i - c_\sigma^i) \cdot f_0) \; \ldots \; ((c_\mu^i + c_\sigma^i) \cdot f_0)]$. This is a window centered on the ratio corresponding the the cluster mean and ranging one standard deviation on either side. Next a Gaussian window w_i^g with the standard deviation of c_i^σ is applied to shape w_i, $\hat{w_i} = w_i^g \cdot w_i$. The largest amplitude value present in this window, $\max(\hat{w_i})$, is extracted as a feature.

Experiments

To evaluate our proposed clustering scheme for feature extraction, we compare two datasets with four different algorithms. We also compare these results with the common MFCC feature set with the same set of algorithms.

Datasets

All note signals are taken from the Real-World Computing (RWC) (Goto et al. 2003) and University of Iowa Musical Instrument Samples (MIS) (Fritts 1997) datasets. These datasets consist of musical instruments playing scales. For our experiments we used the set of 17 instruments common to both datasets, shown in Table 1. For each instrument, the RWC database features two or three different performers, often on instruments by different manufacturers. Both datasets contain notes played at three dynamic levels: *piano*, *mezzo-forte*, and *fortissimo*. These datasets also contain up to three different articulations per instrument. In addition to those shown in Table 2, both datasets contain Marimba (MB) examples performed with hard, medium, and soft mallets. The Guitar (GU) examples are played with both nail and finger in the RWC dataset. All string instruments contain examples of the notes played on each string.

The original sound files are downsampled to a 44.1 kHz sampling rate, 16-bit per sample, single channel waveform. The audio utility *SoX*[1] was used to detect silence and splice the recordings into individual files, each representing an isolated musical note. The datasets are then batch normalized to the range [0,1] using the audio utility *normalize*.[2] Within each dataset and for each instrument, the loudest gain in any of the files is scaled to a value of one and the other files are adjusted accordingly, preserving the relative dynamic levels between instrument examples. We clipped the files to two seconds and added a 10 ms fade-in to the beginning and a 10 ms fade-out to the end of the sample.

Algorithms

For our experiments, we compare several different classifiers common in musical instrument recognition. We demonstrate our feature extraction approach with two Bayesian classifiers, k-nearest neighbors, and a support vector machine.

A Bayesian network is a probabilistic graphical model that represents the conditional dependencies of a set of random variables through a directed acyclic graph, providing a compact representation of joint probability distributions over these variables. A trained classifier can determine the class label of an unseen example that has the highest probability of explaining the values of the example's features.

We use two different Bayesian network structures. Näive Bayes (NB) is a common model that assumes conditional independence between the features, given the class label. In previous work, we have shown the utility of modeling dependencies between frequency features in instrument classification (Donnelly and Sheppard 2013). In our second Bayesian network (BN), each feature a_i is conditionally dependent on the previously feature a_{i-1} as well as the class and are ordered accordingly the ratio window from which they were extracted (Figure 4).

Family	Instrument	RWC	MIS
Brass	French Horn (FH)	655	96
	Trumpet (TR)	607	212
	Trombone (TB)	856	82
	Tuba (TU)	540	71
Woodwind	Flute (FL)	657	227
	Clarinet (CL)	1080	139
	Soprano Sax (SS)	889	192
	Alto Sax (AS)	891	192
	Oboe (OB)	593	104
	Bassoon (BS)	1079	122
String	Violin (VN)	1344	266
	Viola (VA)	1259	241
	Violoncello (VC)	1316	352
	Contrabass (CB)	1385	264
	Guitar (GU)	2817	343
Percussion	Marimba (MB)	871	529
	Piano (PN)	2424	206
Total		**19,263**	**3638**

Table 1: List of 17 instruments common the MIS and RWC datasets and the number of examples in each dataset.

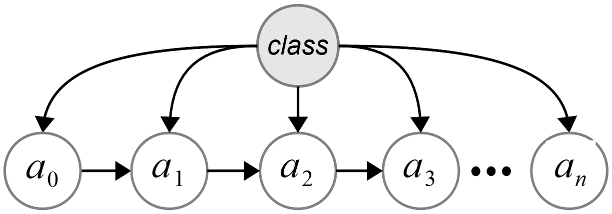

Figure 4: Structure of Bayesian network (BN) with dependencies between frequency features.

The k-nearest neighbors algorithm (k-NN) is a common instance-based learning algorithm that predicts a previously unseen example's class based on the k closest training examples in the feature space. Based on preliminary testing, we use a value of $k = 1$ in all experiments.

A support vector machine (SVM) is a model determines the largest margin separating two classes of data in the feature space and uses this discriminant to classify new examples. We examined several different kernels for this task and we used a polynomial kernel of degree 2 in all experiments.

Experimental Design

Within each dataset, we construct binary datasets for each instrument. For an instrument i with n examples, we use these as the positive examples with class label i. We then choose a random instrument j where $j \neq i$ and choose an

[1] http://sox.sourceforge.net/

[2] http://normalize.nongnu.org/

Instrument	NO	ST	VI	SP	NV	PE
FH	M,R	R				
TR	M,R	R	M,R			
TB	M,R	R	R			
TU	M,R	R				
FL	M,R	R	M,R			
CL	M,R	R	R			
SS	M,R	R	M,R			
AS	M,R	R	M,R			
OB	M,R	R	R			
BS	M,R	R	R			
VN	M,R			R	R	
VA	M,R			R	R	
VC	M,R			R	R	
CB	M,R			R	R	
PI	M,R	R				R

Table 2: List of articulations present in the RWC (R) and MIS (M) datasets. **NO** = Normal, **ST** = Staccato, **VI** = Vibrato, **SP** = Spiccato, **NV** = No Vibrato, and **PE** = Pedal.

example from this instrument at random. We repeat this process until we identify n negative examples with class label not_i. Both the positive and negative features are extracted using the cluster signature for instrument i. Each binary dataset is equally weighted with positive and negative examples, but the size of each BR dataset varies by instrument.

We use 10-fold cross-validation to evaluate when training and testing on the same dataset (M/M, R/R). In the cross dataset experiments (M/R, R/M), we train models on one dataset and test on the other.

For performance evaluation, we report the F-measure which reflects a weighted average of the precision and recall. This measure gives a sense of accuracy of each binary class label individually. For each binary classifier, we take the weighted F-measure between the two classes and then average the F-Measures over the set of binary classifiers.

Results

In Table 3 we show the F-measure for each binary classifier by instrument for the feature set of amplitude values extracted by a signature with 30 trained clusters. We also evaluate across our two datasets MIS and RWC using three different feature sets. We trained our signature models with 20 and 30 clusters, extracted features, and tested these features on four algorithms. We chose 20 and 30 clusters to compare with a baseline feature set, and we will tune the number of clusters in future work.

For comparison, we tested using the common feature set of the mean and standard deviation of the first 13 linear MFCCs, a total of 26 features (Table 4). Our preset network structure for the BN classifier is not extensible to the MFCC feature set and was omitted.

MFCC features are commonly used in recognizing single instrument and as expected, outperformed our approach on all algorithms. For space limitations we cannot report all results, but on all datasets and all instruments, our feature set approach scored better than chance, including in the cross dataset evaluations, using only a näive feature set of only amplitude values extracted at strategic locations in the spectra.

Instr.	NB	BN	k-NN	SVM
FH	0.63	0.65	0.67	0.65
TR	0.70	0.70	0.63	0.72
TB	0.65	0.70	0.63	0.72
TU	0.79	0.87	0.80	0.86
FL	0.74	0.77	0.66	0.69
CL	0.69	0.74	0.62	0.59
SS	0.71	0.73	0.61	0.65
AS	0.68	0.70	0.61	0.60
OB	0.72	0.76	0.61	0.72
BS	0.63	0.66	0.74	0.63
VN	0.74	0.77	0.69	0.75
VA	0.72	0.72	0.65	0.62
VC	0.68	0.73	0.72	0.70
CB	0.80	0.84	0.81	0.83
GU	0.80	0.85	0.90	0.85
MB	0.83	0.84	0.87	0.84
PN	0.76	0.81	0.87	0.79

Table 3: Results of the RWC dataset with 30 features showing the F-Measure for each binary classifier.

Features	Algorithm	M	M	R	R
		M	R	R	M
20 Clusters	NB	0.75	0.64	0.76	0.65
	BN	0.75	0.63	0.77	0.67
	k-NN	0.71	0.61	0.72	0.63
	SVM	0.71	0.64	0.72	0.67
30 Clusters	NB	0.73	0.63	0.72	0.64
	BN	0.75	0.64	0.76	0.66
	k-NN	0.72	0.61	0.71	0.63
	SVM	0.70	0.66	0.72	0.68
MFCC	NB	0.86	0.58	0.80	0.71
	k-NN	0.91	0.71	0.91	0.80
	SVM	0.90	0.66	0.86	0.76

Table 4: Results of cross dataset experiments showing the F-Measure averaged over the set of binary classifiers. The top row shows the training set, M or R, and the bottom row indicates the test set.

Discussion

Instead of using a single feature extraction scheme for all examples, we proposed a strategy of training models for feature extraction using prior expectation of regions of importance in an instrument's timbre. We designed this approach with the goal of extensibility to multilabel classification of polyphonic mixtures.

We will extend this approach to multilabel classification in the following manner. Using only single instrument training data, we extract partials, train our signatures, extract amplitude features from the training data, and train and save a binary classifier for each instrument. Given a mixture to label, we will extract the spectral peaks that exceed our frequency-dependent threshold. For each instrument, we hypothesize each peak in the mixture as f_0, calculate the windows based on the ratios of the spectral signature, and extract amplitude features in those locations. For each hypothesized instrument and f_0 value, we will query the relevant binary classifier for a probability of the presence of the instrument in the signal. Repeating the process for all instrument, we return the set of labels of the most probable instruments.

Here, we have demonstrated this approach over large datasets of single instrument tones containing multiple articulations, performers, and dynamic levels. Although the MFCC feature set outperformed our approach, MFCCs are not very robust in the presence of noise and cannot be used as features in polyphonic mixtures without a prior attempt at source separation (Giannoulis and Klapuri 2013), which is a difficult problem in itself. Our approach, however, attempts spectral separation of the sources given known regions of spectral prominence for each instrument. In this work we demonstrated the validity of a feature extraction approach that relies on prior expectations of generalization of an instrument's unique timbre.

Future Work

In preparation to extend this approach to multilabel classification, we will first explore taking other measurements from our signature windows for use as features, experimenting with temporal dependencies, and tuning the number and size of our spectral clusters.

In this work, we allowed the standard deviations of the clusters to grow unbounded. A large standard deviation results in a feature window spanning multiple musical semitones. This is not ideal when extending this technique to polyphonic mixtures as this increases likelihood of source interference. In future work we will modify our k-means implementation to bound the width of the standard deviation to a fixed maximum. Clusters that exceed this standard deviation will be split into two different clusters, allowing a variable number of clusters to be learned for each instrumental signature.

Presently, we use a fixed number of clusters and use all learned clusters as locations for feature extraction. As illustrated in Figure 3, we also learn the density information for each cluster. In bounding the standard deviation, but allowing a variable number clusters, we will prune the cluster set using only the clusters with the highest densities as feature extraction locations. Finally, we will test statistical similarity between signatures learned by different instruments to identify and prioritize cluster locations particularly unique to a specific instrument.

References

Bottou, L., and Bengio, Y. 1995. Convergence properties of the k-means algorithms. In *Advances in Neural Information Processing Systems 7*, 585–592.

Burred, J. J.; Robel, A.; and Sikora, T. 2010. Dynamic spectral envelope modeling for timbre analysis of musical instrument sounds. *Audio, Speech, and Language Processing, IEEE Transactions on* 18(3):663–674.

Deng, J. D.; Simmermacher, C.; and Cranefield, S. 2008. A study on feature analysis for musical instrument classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 38(2):429–438.

Donnelly, P. J., and Sheppard, J. W. 2013. Classification of musical timbre using bayesian networks. *Computer Music Journal* 37(4):70–86.

Essid, S.; Richard, G.; and David, B. 2006. Instrument recognition in polyphonic music based on automatic taxonomies. *Audio, Speech, and Language Processing, IEEE Transactions on* 14(1):68–80.

Every, M. R., and Szymanski, J. E. 2006. Separation of synchronous pitched notes by spectral filtering of harmonics. *Audio, Speech, and Language Processing, IEEE Transactions on* 14(5):1845–1856.

Fritts, L. 1997. The University of Iowa Electronic Music Studios musical instrument samples. *[Online] Available: http://theremin.music.uiowa.edu/MIS.html*.

Fuhrmann, F.; Haro, M.; and Herrera, P. 2009. Scalability, generality and temporal aspects in automatic recognition of predominant musical instruments in polyphonic music. In *Proceeding of the International Symposium on Music Information Retrieval (ISMIR)*, 321–326.

Giannoulis, D., and Klapuri, A. 2013. Musical instrument recognition in polyphonic audio using missing feature approach. *IEEE transactions on audio, speech, and language processing* 21(9-10):1805–1817.

Goto, M.; Hashiguchi, H.; Nishimura, T.; and Oka, R. 2003. RWC music database: Music genre database and musical instrument sound database. In *ISMIR*, volume 3, 229–230.

Herrera-Boyer, P.; Peeters, G.; and Dubnov, S. 2003. Automatic classification of musical instrument sounds. *Journal of New Music Research* 32(1):3–21.

Kitahara, T.; Goto, M.; Komatani, K.; Ogata, T.; and Okuno, H. G. 2007. Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps. *EURASIP Journal on Applied Signal Processing* 2007(1):155–155.

Leveau, P.; Vincent, E.; Richard, G.; and Daudet, L. 2008. Instrument-specific harmonic atoms for mid-level music representation. *Audio, Speech, and Language Processing, IEEE Transactions on* 16(1):116–128.

Livshin, A., and Rodet, X. 2003. The importance of cross database evaluation in sound classification. In *Proceedings of the International Symposium on Music Information Retrieval*.

Luaces, O.; Díez, J.; Barranquero, J.; del Coz, J. J.; and Bahamonde, A. 2012. Binary relevance efficacy for multilabel classification. *Progress in Artificial Intelligence* 1(4):303–313.

Martins, L. G.; Burred, J. J.; Tzanetakis, G.; and Lagrange, M. 2007. Polyphonic instrument recognition using spectral clustering. In *ISMIR*, 213–218.