2015 IEEE 15th International Conference on Data Mining Workshops

Cross-Dataset Validation of Feature Sets in Musical Instrument Classification

Patrick J. Donnelly and John W. Sheppard
Department of Computer Science
Montana State University
Bozeman, MT 59715
{patrick.donnelly2, sheppard}@cs.montana.edu

***Abstract*—Automatically identifying the musical instruments present in audio recordings is a complex and difficult task. Although the focus has recently shifted to identifying instruments in a polyphonic setting, the task of identifying solo instruments has not been solved. Most empirical studies recognizing musical instruments use only a single dataset in the experiments, despite evidence that mapproaches do not generalize from one dataset to another dataset. In this work, we present a method for data-driven learning of spectral filters for use in feature extraction from audio recordings of solo musical instruments and discuss the extensibility of this approach to polyphonic mixtures of instruments. We examine four datasets of musical instrument sounds that have 13 instruments in common. We demonstrate cross-dataset validation by showing that a feature extraction scheme learned from one dataset can be used successfully for feature extraction and classification on another dataset.**

***Keywords*—binary relevance classification, classification, cross-dataset validation, feature extraction, k-nearest neighbor, instrument recognition, machine learning, music, music information retrieval, musical note separation, timbre**

I. Introduction

Musical instrument recognition is an important research task to the area of music information retrieval (MIR). Many studies have explored recognizing individual musical instruments in isolation. However, these approaches are often sensitive to feature input and classification algorithms and do not generalize between different datasets.

Livshin and Rodet demonstrated that many approaches to musical instrument classification do not generalize from one dataset to another [1]. Using five datasets and seven instruments, the authors performed cross-dataset evaluations and discovered accuracies of 20% to 60% when training on one dataset and testing on another, despite classification results of over 90% for any single dataset using cross-validation. This indicates that the models learned on a single dataset tend overfit and are not extensible to other datasets.

Recent work in the field has shifted to the more complex case of identifying the instruments present in polyphonic mixtures. This is a more difficult problem because the spectral content of the constituent tones can overlap in time and frequency. Most of the approaches developed to recognize individual instruments are not scalable to the more complex case of polyphonic instrument mixtures. [2].

In this paper, we propose a binary relevance feature extraction technique for identifying solo instruments that is designed to be extensible to recognizing instruments in polyphonic mixtures. We demonstrate a data-driven approach to learn areas of prominent harmonics for each instrument and use these resulting signatures to inform the feature extraction stage, described in Section III. In Section IV, we describe a feature set representing energy values extracted only from these regions of prominence learned for each instrument. We normalize the amplitude features by the amplitude of the fundamental frequency, which better enables comparison of features extracted from two different datasets. Using these instrument-specific features, we evaluate this approach in a series of binary relevance classification experiments. In Section V, we validate our approach by showing the ability to use an instrument's signature learned from one dataset to extract features from a different dataset. Lastly, we demonstrate the generalizability of this approach using 13 musical instruments and cross-validation across four different datasets.

II. Related Work

For the task of recognizing isolated instrument tones, researchers have attempted a variety of feature extraction schemes (see [3] for a review) and classification algorithms (see [4] for a review). For the more complicated task of instrument recognition within polyphonic mixtures, there have been several general approaches. The first general approach considers the mixture as a whole, extracting general features directly without attempting any source separation [5], [6], [7], [8]. Many approaches require knowledge of the fundamental frequency, onset time, and duration [6] – information that will not be readily available for real-word data. Others require training on every possible combination of instruments [5], [7], an approach that is not extensible to unseen combinations of instruments and is not feasible for a large number of instruments.

The second approach to classifying mixtures is to adapt existing algorithms to perform multilabel classification directly. Researchers have attempted a multilabel multi-layer perceptron [9], [10], hidden Markov model [11], multilabel decision tree [12], and multilabel k-nearest neighbor [12].

The third and most common approach is the estimation of source separation and classification of the sounds individually. Approaches include matching single instrument templates

978-1-4673-8493-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDMW.2015.213

IEEE computer society

within a mixture [13], selecting features that minimize interference between sources [14], [15], and modeling a decomposition of the signal mixtures [16]. Our approach is designed for the estimation of source separation.

Many of these approaches have significant limitations, such as the use of very few examples or the use of only hand-picked instruments [14], [9], [17], low accuracy results [18], [19], or inability to scale to a previously unseen instrument combination [5], [7] (see [20] for a discussion). We know of only one study [21] that addresses cross-dataset validation on dataset with a single, non-comprehensive experiment.

III. Learning Spectral Filters

In music the harmonic partials of individual tones are interleaved in both the frequency and time domains. In some cases, partials from multiple instruments will overlap, causing destructive or constructive interference. This section describes our data-driven approach to training instrument specific spectral filters for use in feature extraction. Appendix A walks through a detailed example of this procedure.

A. Signal Processing

First we transform the audio signals to the frequency domain using a Fast Fourier Transform (FFT) with a single time window. The resulting amplitudes are scaled by $10 \cdot \log 10$ dB to a power/frequency scale. Since the amplitudes of harmonics in the higher frequencies fall off rapidly relative to the amplitude of the fundamental, working with log amplitudes preserves the importance of the harmonics relative to nearby frequencies.

B. Peak Extraction

For each instrument signal, we seek to extract the harmonics in the spectra. To accomplish this, we establish a threshold above the noise floor and identify any peaks whose amplitudes exceed the threshold (see Figure 4 in Appendix A). We employ a sliding frequency-dependent threshold proposed by [22] and discussed in [23]. This approach permits identifying peaks as significant to their local frequency neighborhood, allowing the capture of significant peaks even in the higher frequency range.

Next, we identify the fundamental frequency f_0 in the signal. Since we examine signals containing only one instrument, we assume the fundamental is the significant peak with the lowest frequency. We extract the frequency location of this peak within a localized window of 32 samples. Using a small window allows capturing the maximum peak in the frequency neighborhood, rather than a local maximum corresponding to a side-lobe, such as those shown in Figure 1.

We extract any amplitude peaks that exceed the threshold and note the frequency location of each peak. In this stage, we are concerned with locating each significant peak relative to f_0. For each peak p in the signal, we save a ratio r calculated as $r = p \div f_0$. We repeat this process for all files for each instrument and save the ratios in a single-dimension vector, with duplicate values allowed. By capturing the ratio to fundamental rather than absolute frequency values, we can normalize away the pitch of the note, allowing direct comparisons between notes with different pitches.

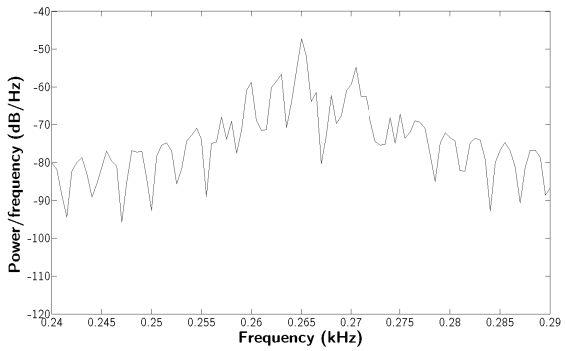


Fig. 1: Zoomed view of the fundamental frequency of a Trumpet playing 265 Hz. The highest peak represents f_0 and the other local peaks are side-lobes resulting from the FFT.

C. Clustering

We then cluster the vector of ratio data to learn the locations of various harmonics important to each instrument. We use the common k-means clustering algorithm [24] to partition the set of ratios into a set of Gaussian clusters. For each cluster, we note the mean and standard deviation and save this set as the instrument's spectral signature. This signature is used to extract features for the classification experiments.

We begin with an initial k=10 clusters. Since musical instruments contain a quasi-harmonic pattern of partials at near integer ratios, we seed the initial k clusters with integer values $[2 \ldots k+1]$, corresponding to the first ten overtones above f_0. We modify the traditional k-means to permit changing the number of clusters as the algorithm progresses. At each iteration, if the standard deviation exceeds 0.5, representing half the distance between two harmonics, the cluster is split into two different clusters. Likewise, if the means of two clusters overlap by less than $\sigma = 0.5$, thy are combined into one. This method yields a variable number of clusters for each instrument and dataset (see Table I). Although the majority of the ratios learned are near-integer ratios, many clusters learned center around inharmonic ratios (e.g., $\mu = 11.57$). Using these clusters, we return a spectral signature for each instrument and dataset. In the feature extraction stage of our experiments, the instrument signature is applied as a spectral mask. Only the spectral energy underneath the signature will be considered for feature extraction while the rest of the spectral signal is disregarded as noise.

IV. Experiments

To evaluate our proposed feature extraction scheme, we perform several classification experiments. In the first, we show that an instrument signature learned from one dataset can be used to extract features on another dataset. In the second, we demonstrate cross-dataset validation by training our models on one dataset and testing on another dataset.

A. Dataset Sources

For our experiments we select the set of 13 instruments common to four different datasets, shown in Table II. The

TABLE I: List of number of clusters learned by instrument and dataset.

Instrument	MUMS	MIS	RWC	PHO
FrenchHorn	29	60	54	88
Trumpet	47	26	44	83
Trombone	64	86	95	110
Tuba	55	120	71	87
Flute	13	38	87	106
Clarinet	39	107	56	82
AltoSaxophone	52	75	84	94
Oboe	30	43	39	30
Bassoon	82	86	96	100
Violin	52	100	71	56
Viola	59	80	72	120
Cello	87	86	102	102
Contrabass	94	100	94	108

TABLE II: List of 13 instruments common to the four datasets and the number of examples in each dataset.

Family	Instrument	MUMS	MIS	RWC	PHO
Brass	French Horn (FH)	37	55	327	531
	Trumpet (TR)	66	106	210	416
	Trombone (TB)	36	82	286	748
	Tuba (TU)	32	65	270	813
Woodwind	Flute (FL)	37	113	221	764
	Clarinet (CL)	37	139	359	752
	Alto Sax (AS)	14	97	297	607
	Oboe (OB)	32	91	198	530
	Bassoon (BS)	32	64	360	632
String	Violin (VN)	45	832	575	572
	Viola (VA)	42	583	542	706
	Violoncello (VC)	47	658	565	743
	Contrabass (CB)	44	617	551	766
	Total	**501**	**3502**	**4761**	**8580**

McGill University Master Samples (MUMS) is a collection of instrument samples, published on compact discs between 1987-1989 [25]. The University of Iowa Musical Instrument Samples (MIS) dataset was created by the Electronic Music Studios at the University of Iowa in 1997 [26]. The Real World Computing (RWC) Music Database is a large-scale music database created specifically for research purposes in 2003 by Japan's National Institute of Advanced Industrial Science and Technology [27]. The Philharmonia Orchestra Sound Sample Collection (PHO) is collection of recordings of various musical instruments created by London's Philharmonia Orchestra, freely available on their website [28].

The datasets range in size from small, containing a few dozen examples of each instrument (MUMS, MIS), to large, containing hundreds of examples of each instrument (RWC, PHO). The datasets are CD quality sound or better, with the exception of the PHO dataset which is in MP3 format.

B. Preprocessing

These datasets consist of musical instruments systematically playing chromatic scales. The MIS, RWC, and PHO datasets contain examples at three different dynamic levels. The smaller MUMS dataset contains examples only at a medium (*mezzo-forte*) dynamic level. All original sound files are downsampled to a 44.1 kHz sampling rate with 16-bit per sample, and mixed down to a single channel waveform. For the lower quality PHO dataset the audio was upsampled to the aforementioned compact disc quality.

We used the *SoX*[1] audio tool to split the musical scales into individual files, each containing a single musical note. Since the frequency resolution of the FFT depends on the number of input samples in the time domain [29], we set all files to be one second in length. If the musical note is shorter than one second, silence was added to lengthen the file. We did not interpolate or repeat the signal to avoid creating any artificial spectral artifacts. If the musical note sample is longer than one second, the file was trimmed to one second. A fade-in and a fade-out of 10 milliseconds each was imposed to eliminate any discontinuities in the waveform resulting from the previous step. For each instrument within each dataset, the audio files were normalized in amplitude relative to loudest gain in any single file. This process preserves the relative dynamic levels between examples for each instrument within each dataset.

C. Feature Extraction

For each example, we first convert the sound file to spectral domain using an FFT as described in Section III-A. For each instrument and each dataset, we use the signatures learned in Section III-C as spectral filters in order to extract amplitude features for use in the classification experiments. For each example, the fundamental frequency is identified as described in Section III-B. Next, the instrument signature is applied to the amplitude spectra as a spectral mask. Each cluster c of the signature has a mean c_μ and a standard deviation c_σ.

For each Gaussian cluster in the signature, we calculate a window centered on the ratio corresponding to the cluster mean and ranging plus and minus one standard deviation. The ratio is calculated relative to f_0 and each window ranges $((c_\mu - c_\sigma) \cdot f_0)$ to $((c_\mu + c_\sigma) \cdot f_0)$. Within each cluster window, the maximum amplitude is extracted as a feature. This is repeated for each Gaussian cluster in the signature. In these experiments we use the very simple feature of the maximum amplitude value within each window. In future work, we will explore using other more complex spectral features, such as those described in [3]. As our goal in this work centers on demonstrating cross-dataset validation, we avoid potentially overfitting individual datasets by optimizing from a complex set of spectral features, as is common in the literature, and instead demonstrate our approach using a simple feature space.

Lastly, we normalized these amplitude values relative to the amplitude of f_0. Considering feature values relative to f_0 allows us to compare notes played at different dynamic levels. Furthermore, it also permits comparing notes between datasets, helping to overcome differences caused by the recording procedures of the individual datasets. In other words, this allows comparison of features extracted from the same instrument but from different datasets. Appendix B walks through a detailed example of this feature extraction procedure.

D. Experimental Design

Binary relevance (BR) classification is a common decomposition approach to multilabel classification. In BR classification, a separate classifier is trained for each class label and this binary classifier is responsible for determining if the label as relevant or irrelevant to each example [30]. The BR approach to multilabel classification is often the baseline

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

against which other multilabel classification approaches are compared experimentally [31]. Many approaches to multilabel classification increase in complexity as the number of class labels increase and are not scalable to a large number of labels. BR classification, on the hand, scales linearly in the number of models as the number of class labels increases. Another key assumption in BR classification is the independence of class labels. In many domains, multilabel data contains dependency between labels and researchers are exploring approaches to multilabel classification that can exploit dependencies between class labels [32], [33].

In this work, we train a BR classifier for each musical instrument that determines if that instrument is present or not present in the signal. Although the experiments reported in this work examine both training and testing on datasets of solo instruments, our BR approach is designed to extend naturally to multilabel classification, training on datasets containing only solo instruments but permitting testing on signals containing polyphonic mixtures. The is a key contribution that differentiates our approach from other studies on solo instrument classification. Compared to other approaches to multilabel classification of polyphonic mixtures, such as [5], our BR approach is extensible to new instruments, requiring only solo examples of the new instruments for training.

To train and test our instrument-specific BR classifier, we organize our datasets into binary datasets for each individual instrument. For each instrument i, we create a dataset D_i in which 50% of the dataset are examples of instrument i, assigned the positive class label (+) . The other 50% of the examples in the dataset are examples of other instruments, any instrument $\neg i$, which is assigned the negative class label $(-)$. To select examples for the negative class label, we randomly select one of the other twelve instruments and randomly select with replacement a sound example of the chosen instrument. Each dataset D_i contains an equal number of positive and negative class labels. Since the number of examples of instrument i available differs between instruments and datasets, the total size of each dataset D_i is twice the value given in Table II. For both positive and negative examples, features are extracted using the cluster signature (see Section III-C) for the positive instrument class.

Throughout this work, we use the terms self-classification and cross-dataset classification to describe two different experimental designs. For self-classification tasks, we train and test on the same dataset, using a 10-fold cross-validation approach, reporting the average of the results of the 10-folds. For the cross-dataset classification tasks, we train one dataset and test on a different dataset. In our signature validation experiments (Section V-A), we the use self-classification paradigm. In the cross-dataset experiments (Section V-B), we report the results using the cross-dataset approach for all training and test set combinations from the four datasets. For the case when the training and test sets are the same, we use the self-classification experimental design.

k-NN is a common non-parametric lazy learning algorithm used for classification. For any unseen example, k-NN predicts the class label by finding the k nearest examples from the training set that minimize a distance metric. From that set of k neighbors, the class label of the majority of neighbors is assigned to unseen example [34]. Based on preliminary

TABLE III: Results of the signature validation experiments showing the F-measure for each binary classifier (instrument) for each dataset. Figures IIIa − IIId report the results using instrument signatures learned from each of the four different datasets, respectively. The italicized results indicate the signature was learned from the same dataset that is tested.

(a) Signature learned from the MUMS dataset.

Instrument	MUMS	MIS	RWC	PHO
French Horn	*0.64*	0.70	0.64	0.76
Trumpet	*0.75*	0.63	0.82	0.73
Trombone	*0.51*	0.58	0.67	0.64
Tuba	*0.65*	0.65	0.58	0.81
Flute	*0.77*	0.75	0.71	0.67
Clarinet	*0.73*	0.57	0.78	0.71
Alto Saxophone	*0.53*	0.61	0.61	0.93
Oboe	*0.54*	0.72	0.72	0.50
Bassoon	*0.73*	0.69	0.74	0.74
Violin	*0.72*	0.61	0.58	0.63
Viola	*0.71*	0.53	0.70	0.52
Cello	*0.79*	0.73	0.75	0.73
Contrabass	*0.80*	0.89	0.79	0.84

(b) Signature learned from the MIS dataset.

Instrument	MUMS	MIS	RWC	PHO
French Horn	0.74	*0.75*	0.81	0.74
Trumpet	0.88	*0.91*	0.83	0.80
Trombone	0.72	*0.74*	0.72	0.68
Tuba	0.77	*0.88*	0.87	0.90
Flute	0.69	*0.73*	0.68	0.72
Clarinet	0.85	*0.87*	0.87	0.89
Alto Saxophone	0.75	*0.79*	0.76	0.75
Oboe	0.78	*0.83*	0.74	0.78
Bassoon	0.70	*0.67*	0.76	0.68
Violin	0.86	*0.87*	0.88	0.86
Viola	0.74	*0.74*	0.73	0.70
Cello	0.78	*0.78*	0.77	0.80
Contrabass	0.89	*0.89*	0.90	0.87

(c) Signature learned from the RWC dataset.

Instrument	MUMS	MIS	RWC	PHO
French Horn	0.71	0.75	*0.77*	0.78
Trumpet	0.71	0.69	*0.73*	0.71
Trombone	0.75	0.76	*0.76*	0.74
Tuba	0.88	0.91	*0.86*	0.90
Flute	0.78	0.77	*0.77*	0.75
Clarinet	0.90	0.86	*0.88*	0.88
Alto Saxophone	0.75	0.78	*0.74*	0.77
Oboe	0.81	0.83	*0.80*	0.80
Bassoon	0.87	0.85	*0.87*	0.85
Violin	0.86	0.84	*0.84*	0.86
Viola	0.78	0.77	*0.78*	0.75
Cello	0.85	0.84	*0.83*	0.83
Contrabass	0.92	0.92	*0.91*	0.91

(d) Signature learned from the PHO dataset.

Instrument	MUMS	MIS	RWC	PHO
French Horn	0.81	0.79	0.80	*0.80*
Trumpet	0.76	0.78	0.76	*0.77*
Trombone	0.76	0.78	0.76	*0.77*
Tuba	0.91	0.92	0.89	*0.90*
Flute	0.80	0.87	0.84	*0.82*
Clarinet	0.91	0.87	0.89	*0.87*
Alto Saxophone	0.76	0.74	0.75	*0.75*
Oboe	0.89	0.88	0.88	*0.87*
Bassoon	0.87	0.86	0.86	*0.85*
Violin	0.82	0.85	0.85	*0.82*
Viola	0.82	0.81	0.83	*0.82*
Cello	0.83	0.82	0.80	*0.80*
Contrabass	0.90	0.90	0.92	*0.90*

testing, we use $k = 7$ in our experiments and the Euclidean distance metric, commonly used with continuous variables. In the information retrieval domain, the metric known as precision is the fraction of retrieved examples that are relevant and the metric recall is the fraction of all relevant documents retrieved. To evaluate the performance of our experiments, we report the F-measure, a weighted averaged of precision and recall.

V. Results

A. Signature Validation

In this experiment, we explore the generalizability of our feature extraction approach. We demonstrate that an instrument signature learned from one dataset can be used for feature extraction for the same instrument in a different dataset. In these signature validation experiments, we use the self-classification paradigm described in Section IV-D.

For each dataset, we consider the cluster signature learned for each instrument. This signature informs the locations in the signal of the features to extract. We apply this signature to each of the other datasets and extract the relevant features. In other words, we use the locations of the features learned for one instrument in one dataset to extract the features for the same instrument from another dataset.

In Table III we report the F-measure result of each binary classifier. For most instruments and datasets, we show that a signature learned from one dataset can be successfully applied for feature extraction on another dataset. In numerous cases, we found a higher accuracy when applying a signature from one dataset to another dataset. For example, many of the instrument signatures learned from the large, high quality RWC dataset (Table IIIc) produced a higher score than the self-classification results of the RWC dataset itself. This result strongly implies that our BR feature extraction technique finds features that generalize an instrument's musical timbre, regardless of the dataset.

B. Cross-Dataset Validation

In the cross-dataset experiments, we examine the ability of our approach to generalize between datasets. For each dataset, we train a separate BR classifier for each instrument. We then use this trained model to classify each of the other datasets. When the training set and test same are the same, we use the cross-validation approach described above. In Table IV we report the F-measure result of each classifier.

In these experiments, we found that we are able to train on features from one dataset and test on features extracted from another dataset. As expected, we observe a reduced classification accuracy for the cross-dataset experiments compared to the self-classification experiments. However these results are far more promising than the cross-dataset results reported in [1], although, given the differing features and classification algorithms, the results of the two approaches are not directly comparable.

Nevertheless, we are able to classify using the cross-dataset paradigm at rates well above chance for almost all datasets and instruments. In our preliminary experiments, we observed that setting a small value of k, such as $k = 1$ substantially increased accuracy on the self-classification experiments but decreased

TABLE IV: Cross-dataset experiments showing the F-measure for each binary classifier (instrument) for each dataset. The column headers show the test dataset. The italicized values indicate self-classification. All others values represent cross-dataset classification.

(a) Classifier trained on the MUMS dataset

Instrument	MUMS	MIS	RWC	PHO
French Horn	*0.66*	0.65	0.59	0.59
Trumpet	*0.79*	0.67	0.65	0.59
Trombone	*0.62*	0.66	0.65	0.61
Tuba	*0.69*	0.50	0.66	0.71
Flute	*0.78*	0.73	0.72	0.63
Clarinet	*0.81*	0.75	0.78	0.8
Alto Saxophone	*0.59*	0.38	0.47	0.44
Oboe	*0.68*	0.67	0.68	0.71
Bassoon	*0.77*	0.72	0.70	0.68
Violin	*0.73*	0.58	0.67	0.67
Viola	*0.68*	0.63	0.66	0.65
Cello	*0.78*	0.62	0.77	0.63
Contrabass	*0.83*	0.74	0.84	0.77

(b) Classifier trained on the MIS dataset

Instrument	MUMS	MIS	RWC	PHO
French Horn	0.65	*0.77*	0.65	0.62
Trumpet	0.61	*0.91*	0.66	0.61
Trombone	0.65	*0.74*	0.69	0.68
Tuba	0.44	*0.88*	0.42	0.54
Flute	0.66	*0.77*	0.74	0.66
Clarinet	0.63	*0.88*	0.83	0.76
Alto Saxophone	0.66	*0.81*	0.64	0.55
Oboe	0.70	*0.85*	0.67	0.69
Bassoon	0.81	*0.77*	0.73	0.68
Violin	0.66	*0.87*	0.75	0.74
Viola	0.69	*0.78*	0.58	0.61
Cello	0.67	*0.80*	0.67	0.66
Contrabass	0.91	*0.90*	0.88	0.81

(c) Classifier trained on the RWC dataset

Instrument	MUMS	MIS	RWC	PHO
French Horn	0.78	0.75	*0.78*	0.67
Trumpet	0.75	0.74	*0.72*	0.64
Trombone	0.78	0.74	*0.76*	0.72
Tuba	0.59	0.36	*0.87*	0.73
Flute	0.77	0.73	*0.78*	0.67
Clarinet	0.78	0.82	*0.89*	0.75
Alto Saxophone	0.78	*0.75*	0.79	0.53
Oboe	0.80	0.79	*0.82*	0.79
Bassoon	0.83	0.81	*0.86*	0.77
Violin	0.72	0.69	*0.87*	0.77
Viola	0.83	0.60	*0.80*	0.61
Cello	0.88	0.67	*0.84*	0.70
Contrabass	0.93	*0.85*	0.92	0.81

(d) Classifier trained on the PHO dataset

Instrument	MUMS	MIS	RWC	PHO
French Horn	0.70	0.65	0.66	*0.82*
Trumpet	0.62	0.83	0.68	*0.79*
Trombone	0.78	0.73	0.66	*0.79*
Tuba	0.86	0.67	0.83	*0.91*
Flute	0.49	0.74	0.68	*0.85*
Clarinet	0.85	0.80	0.79	*0.88*
Alto Saxophone	0.57	0.55	0.62	*0.79*
Oboe	0.76	0.76	0.81	*0.88*
Bassoon	0.70	0.73	0.77	*0.87*
Violin	0.78	0.66	0.75	*0.84*
Viola	0.68	0.63	0.65	*0.84*
Cello	0.74	0.63	0.76	*0.83*
Contrabass	0.90	0.87	0.87	*0.91*

accuracy on the cross-dataset experiments. This is an example of overfitting to a specific dataset, which is a common problem in the instrument classification literature. As we increased the value of k, the self-classification results decreased as the cross-dataset accuracy increased. In other words, comparing an unknown example to the single nearest instance is useful in the self-classification task, but more neighbors are required to better generalize between instruments across datasets.

VI. Discussion

We present an approach to feature extraction for classification of solo musical instruments. We examine four large datasets each containing examples of 13 musical instruments in common. We propose a data-driven learning approach to find regions of spectral prominence for each musical instrument. We use these spectral filters for extracting features from audio recordings of solo instruments. Since we use a BR experimental design, we need not use the same set of features for each instrument class. Instead we use an instrument specific-set of features for each BR classifier. We design this approach specifically to be extensible to multilabel classification of mixtures of multiple instruments.

First, we demonstrate that our BR feature extraction scheme does generalize between datasets as we show that, for each instrument, the important feature locations learned in one dataset can be successfully used to extract features from another dataset. This result implies that we are capturing features relevant to the specific instrument's timbre, rather than features influenced by the recording procedures, such as the microphone, amplitude levels, and other variations between datasets. Secondly, we demonstrate cross-dataset validation by showing that we can train an instrument-specific BR classifier on one dataset, and test the model on another dataset.

In the musical instrument classification literature, most approaches are heavily biased by the training set and cannot be used to classify other datasets [1]. Cross-dataset validation needs to be goal of any approach that hopes to eventually generalize to real-world musical data. Our cross-dataset experiments demonstrate an ability of our approach to provide such a generalization.

VII. Future Work

In ongoing work, we extend this approach to multilabel classification of polyphonic mixtures of instruments. For each dataset, we train models using the approach described in this paper. Using only recordings of solo instruments, we extract partials, train the instrument signatures, extract amplitude features, and train a BR classifier for each instrument.

Next, we create a dataset of polyphonic mixtures of instrument by selecting two or more unique instruments at random and mix them together. Given an audio signal containing a mixture of unknown instruments to classify, we begin by extracting significant spectral peaks that exceed our frequency-dependent amplitude threshold. We must consider each of these significant peaks as a potential fundamental frequency f_0 for each possible musical instrument. Given an individual peak and a hypothesis of a particular instrument i, we apply the spectral signature of that instrument i and extract amplitude features in those locations, ignoring the rest of the signal. We then query the BR classifier for a probability that instrument i is contained in the mixture. We repeat this process for each instrument hypothesis and significant peak and classify the mixture as containing the set of instruments that returned the highest probabilities.

Appendix A
Signature Learning Example

This appendix walks through a detailed example of the signature learning process described in Section III. We begin with a single instrument, the Clarinet. Consider a sound file of a Clarinet playing a single note, as shown in Figure 2. We then transform the signal to the frequency domain using an FFT, as shown in Figure 3.

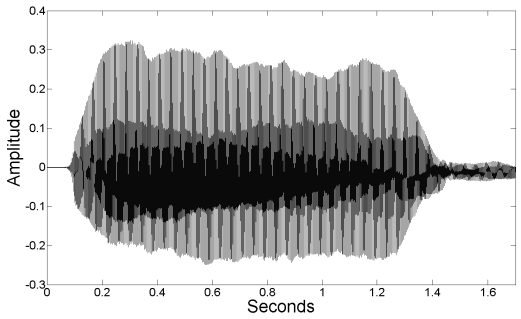


Fig. 2: Waveform of a Clarinet playing middle C (261 Hz)

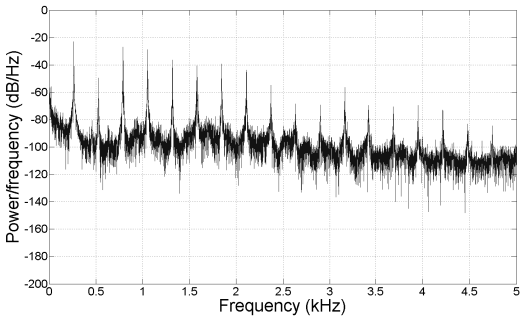


Fig. 3: Spectra of a Clarinet playing middle C (261 Hz)

The next step is to determine the variable-frequency noise threshold as described in Section III-B and shown in Figure 4. We consider any peak above this threshold to be a significant peak. Among those significant peaks, we identify the fundamental frequency f_0 using the procedure described in Section III-B. Our algorithm selects the lowest significant peak, the leftmost peak shown in Figure 3.

After identifying the significant peak threshold, we extract all the locations (in Hertz) corresponding to these peaks. Using the frequency of f_0, we calculate the ratio of the peak to the fundamental. Observe that in the examples shown in Table V there are several significant peaks centering around an integer ratio value. Since we use a single one-second time window in our FFT, we obtain a high frequency resolution and capture the frequency fluctuation over the course of the one second sample.

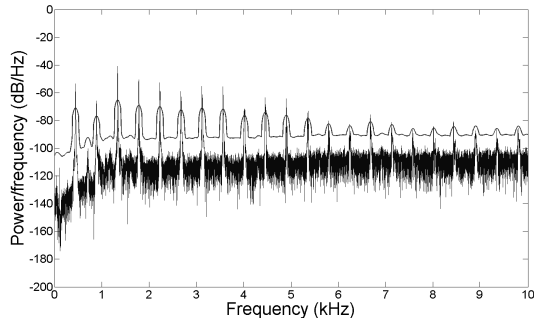


Fig. 4: Amplitude spectrum and threshold a Clarinet note

These values will contribute towards the standard deviation of the signature clusters.

TABLE V: Examples of significant peaks of Clarinet note with $f_0 = 446$

frequency	amplitude	ratio
889.5	−68.88867246	1.994
890.0	−65.80878555	1.996
891.5	−67.97254256	1.999
892.5	−74.96824454	2.001
893.0	−74.00520015	2.002
893.5	−73.05548783	2.003
894.0	−71.43471702	2.004
1333.5	−58.82644509	2.990
1334.0	−57.31556185	2.991
1334.5	−55.70385473	2.992
1335.0	−44.85334492	2.993
1335.5	−40.94094168	2.994
1338.5	−49.02662269	3.001
1340.0	−52.91051806	3.004
… *etc.*		

We repeat this procedure for all other Clarinet sound files in the dataset, such as the simplified examples shown in Table VI. Next, we flatten all these values into a single one-dimensional vector, retaining any duplicate values. At this stage, we do not use any amplitude information but only the ratios corresponding to the frequency locations of the peaks. The energy of the peaks are used in the feature extraction stage of the classification experiments, as described in Section IV-C. For now, we are concerned with learning where to look for significant spectral energy.

TABLE VI: Examples of the ratios extracted from several different Clarinet notes

2.00, 2.99, 4.01, 4.99, 6.03
2.00, 2.97, 3.06, 3.98, 4.95, 5.04, 5.94, 6.07, 6.94, 7.07, 7.92, 8.07, 9.88
2.04, 2.05, 3.04, 3.07, 3.09, 4.05, 4.07, 4.09, 4.12, 5.08, 5.10, 5.12, 5.15
2.00, 2.99, 3.97, 4.97, 5.09, 5.93, 6.96, 7.10, 7.98, 8.98, 9.98, 11.00, 13.13
2.00, 3.01, 3.98, 5.01, 6.01, 6.98
1.97, 2.00, 2.97, 3.01, 3.95, 3.98, 4.01, 5.01, 5.93, 5.97, 6.01
2.00, 3.01, 4.01, 4.98, 6.02, 7.03, 7.98
1.98, 2.03, 3.01, 3.07, 4.01, 4.07, 4.98, 5.04, 5.10, 6.02, 6.11, 7.01
… *etc.*

Next we apply the k-means clustering algorithm on the set of ratio values as described in Section III-C. We then extract the resulting k clusters as the signature for the Clarinet. Each cluster returns a mean μ and standard deviation σ, which we use as window centered on the ratio plus and minus one standard deviation. A larger standard deviation indicates more fluctuation in frequency over the duration of the sound file. For example, the signatures of string instruments, such as the Violin, contain on average a larger standard deviations than other instruments. This corresponds to the natural pitch fluctuation, or vibrato, of the instrument.

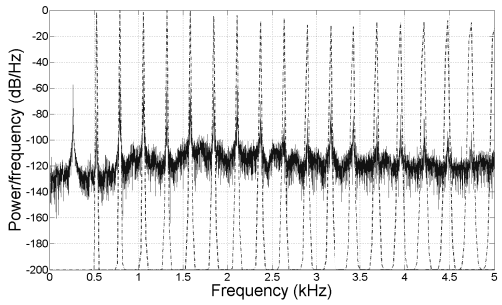
TABLE VII: Example clusters learned for the Clarinet

μ= {2.003, 3.000, 4.006, 4.997, 5.998, 6.988, 7.988, 8.981, 9.984, 10.976 … }
σ= {0.026, 0.037, 0.046, 0.055, 0.056, 0.058, 0.059, 0.064, 0.064, 0.070 … }

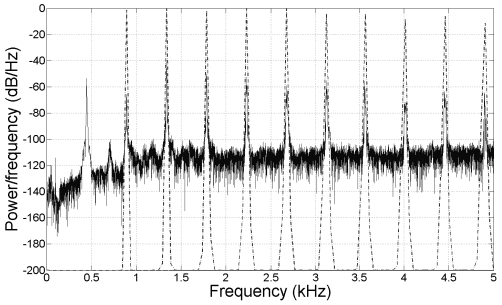
We repeat this procedure for every instrument and for each of the datasets. We learn a unique spectral signature for each instrument and each dataset.

APPENDIX B
FEATURE EXTRACTION EXAMPLE

In this example, we walk through the procedure of extracting features for the classification experiments. For a given instrument hypothesis, using the instrument's learned spectral signature, we extract amplitude features only in regions masked by the spectral filter.



(a) Clarinet playing C (265 Hz) with signature (dashed)



(b) Clarinet playing A (440 Hz) with signature (dashed)

Fig. 5: Examples of signature applied to two different notes

For each example, and for each instrument hypothesis, we use the spectral signature to extract the maximum amplitude

within the window of $\mu_i \pm \sigma_i$ for each i^{th} cluster in the spectral signature. For example,

f_0 is 446.0 Hz
μ_1 = 2.003
σ_1 = 0.026

Calculate window $[881.742, 904.934]$
Extract maximum amplitude in window: -65.81 at 890.0 Hz

This is repeated for all clusters in the signatures (see Table I) and the resulting amplitude values are converted to ratio to the fundamental's amplitude and stored as features, as described in Section IV-C.

Examples of a Clarinet signature applied to the spectra of two Clarinet notes is shown in Figure 5. Since the signatures capture the locations relative to the f_0, we can apply the instrument's signature to any note, regardless of the pitch.

REFERENCES

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

[2] F. Fuhrmann, M. Haro, and P. Herrera, "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*, 2009, pp. 321–326.

[3] J. D. Deng, C. Simmermacher, and S. Cranefield, "A study on feature analysis for musical instrument classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 2, pp. 429–438, 2008.

[4] P. Herrera-Boyer, G. Peeters, and S. Dubnov, "Automatic classification of musical instrument sounds," *Journal of New Music Research*, vol. 32, no. 1, pp. 3–21, 2003.

[5] S. Essid, G. Richard, and B. David, "Instrument recognition in polyphonic music based on automatic taxonomies," vol. 14, no. 1, pp. 68–80, 2006.

[6] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 155–155, 2007.

[7] P. Somerville and A. L. Uitdenbogerd, "Multitimbral musical instrument classification," 2008, pp. 269–274.

[8] P. J. Donnelly and J. W. Sheppard, "Classification of musical timbre using bayesian networks," *Computer Music Journal*, vol. 37, no. 4, pp. 70–86, 2013.

[9] B. Kostek, "Musical instrument classification and duet analysis employing music information retrieval techniques," *Proceedings of the IEEE*, vol. 92, no. 4, pp. 712–729, 2004.

[10] P. Hamel, S. Wood, and D. Eck, "Automatic identification of instrument classes in polyphonic and poly-instrument audio," in *Proceeding of the International Symposium on Music Information Retrieval*, 2009, pp. 399–404.

[11] J. Paulus and A. Klapuri, "Drum sound detection in polyphonic music with hidden markov models," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2009, pp. 14–23, 2009.

[12] W. Jiang, A. Wieczorkowska, and Z. W. Raś, "Music instrument estimation in polyphonic sound based on short-term spectrum match," in *Foundations of Computational Intelligence Volume 2*, 2009, pp. 259–273.

[13] P. Leveau, E. Vincent, G. Richard, and L. Daudet, "Instrument-specific harmonic atoms for mid-level music representation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 116–128, 2008.

[14] J. Eggink and G. J. Brown, "A missing feature approach to instrument identification in polyphonic music."

[15] D. Giannoulis and A. Klapuri, "Musical instrument recognition in polyphonic audio using missing feature approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 9, pp. 1805–1817, 2013.

[16] Y. Hu and G. Liu, "Instrument identification and pitch estimation in multi-timbre polyphonic musical signals based on probabilistic mixture model decomposition," *Journal of Intelligent Information Systems*, vol. 40, no. 1, pp. 141–158, 2013.

[17] J. J. Burred, A. Robel, and T. Sikora, "Polyphonic musical instrument recognition based on a dynamic model of the spectral envelope," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 173–176.

[18] P. Leveau, D. Sodoyer, and L. Daudet, "Automatic instrument recognition in a polyphonic mixture using sparse representations," in *Proceeding of the International Symposium on Music Information Retrieval*, 2007, pp. 233–236.

[19] L. G. Martins, J. J. Burred, G. Tzanetakis, and M. Lagrange, "Polyphonic instrument recognition using spectral clustering," in *Proceeding of the International Symposium on Music Information Retrieval*, 2007, pp. 213–218.

[20] J. G. A. Barbedo and G. Tzanetakis, "Musical instrument classification using individual partials," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 111–122, 2011.

[21] Z. Duan, B. Pardo, and L. Daudet, "A novel cepstral representation for timbre modeling of sound sources in polyphonic mixtures," in *IEEE Conference of Acoustics, Speech and Signal Processing*, 2014, pp. 7495–7499.

[22] M. R. Every and J. E. Szymanski, "Separation of synchronous pitched notes by spectral filtering of harmonics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1845–1856, 2006.

[23] P. Donnelly and J. Sheppard, "Clustering spectral filters for extensible feature extraction in musical instrument classification," in *The Twenty-Seventh International Florida Artificial Intelligence Research Symposium*, pp. 37–42.

[24] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Applied statistics*, pp. 100–108, 1979.

[25] F. Opolko and J. Wapnick, "Mcgill university master samples (MUMS). 11 cd-rom set," *Faculty of Music, McGill University, Montreal, Canada*, 1989.

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

[27] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Music genre database and musical instrument sound database," in *Proceedings of the International Symposium on Music Information Retrieval*, vol. 3, 2003, pp. 229–230.

[28] "Philharmonic orchestra sound sample collection," *[Online] Available: http://www.philharmonia.co.uk/explore/make_music*.

[29] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[30] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data mining and knowledge discovery handbook*, 2010, pp. 667–685.

[31] O. Luaces, J. Díez, J. Barranquero, J. J. del Coz, and A. Bahamonde, "Binary relevance efficacy for multilabel classification," *Progress in Artificial Intelligence*, vol. 1, no. 4, pp. 303–313, 2012.

[32] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, pp. 333–359, 2011.

[33] M.-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 999–1008.

[34] E. Fix and J. L. Hodges Jr, "Discriminatory analysis-nonparametric discrimination: consistency properties," DTIC Document, Tech. Rep., 1951.