Investigating Machine Learning Approaches for Predominant Musical Instrument Recognition in Polyphonic Music

Yosua Muliawan and Jérémie Clos

University of Nottingham
psyym3@nottingham.ac.uk
jeremie.clos@nottingham.ac.uk

Abstract. The automatic recognition of musical instruments has many practical applications, such as automating the tagging of musical instruments in a piece of music for music information retrieval. In this work we investigate the effectiveness of the three key components of a music instrument recognition system: window configuration, feature extraction and classifiers. Our results show that the Hamming window function provides the best accuracy in comparison to the Hann and Blackman-Harris window functions, in combination with a short time window and a Random Forest classifier. We identify several key audio features that provide the best accuracy when classifying musical instruments from polyphonic music.

Keywords: instrument recognition ; music tagging ; machine learning

1 Introduction

Current research in musical instrument recognition is often presented in one of two forms, namely musical instrument recognition in monophonic, or polyphonic music. Studies of instrument recognition in monophonic music deal with isolated recordings of a single instrument. On the other hand, polyphonic musical instrument classification attempts to classify music from a mixture of musical instruments, which are mixed digitally or sampled from real-world music. In this paper, we focus on the classification of music in polyphonic mixtures. Specifically, on the multiclass, single-label classification of musical instruments from polyphonic music. We perform a comparative study of three key aspects of a musical instrument classification system: windowing, feature extraction, and classifiers. We seek to answer the following research questions:

(RQ1) Which combination of window function and length provides the best performance in recognizing the predominant musical instrument from polyphonic music?
(RQ2) Which set of features optimally represents an audio signal in recognizing the predominant musical instrument in polyphonic music?
(RQ3) Which classifier performs best, given its optimal set of features in recognizing the predominant musical instrument from polyphonic music?

2 Related work

Eronen and Klapuri's early work in instrument classification presented an approach using a wide set of features to model the spectral and temporal characteristics of sound (Eronen & Klapuri, 2000). They tested their approach against classifiers such as K-Nearest Neighbors (KNN) and Gaussian models to classify monophonic music. This approach was subsequently the baseline work on musical instrument recognition, as well as one of the earliest works on the subject (Seipel, 2018). Subsequent research expanded on the number of features used to classify instruments and concluded that the Mel-Frequency Cepstral Coefficients (MFCCs) feature gave the overall best accuracy (Eronen, 2001). Work on monophonic instrument recognition culminated with the usage of a Support Vector Machine (SVM) classifier that was trained on Spectro-Temporal Receptive Fields (STRF), which achieved 98.7% accuracy on the RWC dataset (Patil, Pressnitzer, Shamma, & Elhilali, 2012).

Recent advances in deep learning have also allowed for modern techniques such as Convolutional Neural Networks (CNNs) to be employed in the task. Park and Lee investigated learned features from CNNs, which outperformed traditional handcrafted features (such as MFCC) and traditional machine learning classifiers (Park & Lee, 2015). Lostanlen and Cella compared different kernels in deep convolutional networks and obtained the best classification accuracy by hybridizing three convolutional layers into a single architecture (Lostanlen & Cella, 2016). Because we aim to study features, these works are out of the scope of this paper.

In polyphonic musical instrument recognition, Fuhrmann (Fuhrmann et al., 2012) presented a complete overview of the problem of musical instrument classification. In his doctoral thesis, Fuhrmann derived a dataset to classify musical instruments from professionally recorded, real-world, western music. Subsequently, this dataset was improved upon by Bosch et al. (Bosch, Janer, Fuhrmann, & Herrera, 2012), which eventually was formalized into the IRMAS dataset, which we use in this paper.

Recent work in polyphonic musical instrument recognition focuses on finding the optimal set of features that can differentiate different timbral characteristics of musical instruments. For instance, Fuhrmann et al. investigated the optimal features derived from many domains of parameterization (Fuhrmann, Haro, & Herrera, 2009), in which they found that correlation-based feature selection mostly selected barkband and cepstral-based features. Slizovskaia et al. investigated a very large set of features from the CUIDADO project and found that the XGBoost algorithm significantly outperformed the traditional SVM approach (Slizovskaia, Gómez Gutiérrez, & Haro Ortega, 2016).

More recently, Gururani et al presented an attention mechanism for polyphonic music (Gururani, Sharma, & Lerch, 2019). They used a traditional fully connected neural network as well as a recurrent neural network (RCNN) to analyze the performance of the proposed attention mechanism, and showed an improvement in the accuracy of the models. Lastly, Han et al. presented an approach using deep convolutional neural networks to recognize instruments from polyphonic music, using the Mel-spectrogram of the audio as input to the network (Y. Han, Kim, & Lee, 2016).

3 Experimental design

In this section we discuss the design of the three key components explored in this paper: window configuration, audio features, and classifiers.

Audio features Feature extraction is the process of extracting underlying information about an audio excerpt that can be used to characterize a particular piece of audio. In this process, data is encoded into a smaller, more compact representation that is more manageable for a machine learning algorithm to process. Recent research into audio features have produced approaches beyond the classic domains of parameterization (temporal, frequency, and cepstral), such as the wavelet domain, and bio-inspired approaches (Alías, Socoró, & Sevillano, 2016). However, the classic domains remains prevalent in the state of the art musical instrument recognition systems (Herrera-Boyer, Peeters, & Dubnov, 2003). We list the features used in the summary table 1.

List of window functions The window function greatly determines the quality of features from the spectral and cepstral domain, but are not applicable to the temporal features. It is applied to data after the data has been segmented into fixed-length blocks. Window functions are used to mitigate a phenomenon known as spectral leakage. We list the window functions that are explored in this paper, with input signal n, and window length N, in table 2.

Window length and hop size The window length refers to the length of the fixed-size block where the signal is taken and converted into features. The hop size refers to the distance between a window, and the following consecutive window. The window length determines the resolution of a given window (how much of the audio excerpt is represented by the window). On the other hand, the hop size determines the amount of overlap each window has. For simplicity, this paper uses a 50% window overlap.

We investigate a short window length of 46ms, inspired by its usage in (Slizovskaia et al., 2016), a medium window length of 500ms, and a long window length of 1s as mentioned in (Alías et al., 2016). Window configurations that will be explored in this project are listed in table 3. With a sampling rate of 44.1 KHz, each window length described above is converted into the number of samples in audio.

Classifiers As the final component of a musical instrument recognition system, three types of classifiers are explored in this paper: K-Nearest Neighbors, Random Forest, and Support Vector Machine.

Table 1. List of features

Domain	Feature	Description
Temporal	Log attack time	Duration of perceptual audibility of sound. Usages mentioned in (Herrera-Boyer et al., 2003).
	Zero-crossing rate	Number of sign changes of an audio signOkal divided by the number of samples. Used in (Benetos, Kotti, & Kotropoulos, 2006).
Spectral	Spectral flatness	Uniformity of the frequency distribution of the power spectrum.
	Spectral Roll-off	roll-off frequency in which 85% of the total energy contained in the spectrum lies.
	Spectral flux	Speed of change in consecutive spectrums.
	Spectral centroid	Center of gravity of the spectral energy.
	Spectral variance	Spread of spectral energies.
	Spectral skewness	Degree of symmetry of the distribution of the spectral energy.
	Spectral kurtosis	Degree of sharpness of the distribution of the spectral energy.
Cepstral	MFCC	Input is Fourier transformed then fed through the Mel scale filter bank, log-scaled, and decorrelated using the Discrete Cosine Transform (DCT). Used in (Lyon, 2010; Benetos et al., 2006; Fuhrmann et al., 2012, 2009).
	BFCC	Input is Fourier transformed and then fed to a bark scale filter bank, log-scaled and then decorrelated using DCT. Used in (Brent, 2009; Gulzar, Singh, & Sharma, 2014).

Table 2. Window functions

Function	Definition
Hann	$w(n) = 0.5 - 0.5 \cdot \cos(\frac{2\pi n}{N})$ with $0 \leq n \leq N$
Hamming	$w(n) = 0.54 - 0.46 \cdot \cos(\frac{2\pi n}{N})$ with $0 \leq n \leq N$
Blackman-Harris	$w(n) = 0.35875 - 0.48829 \cdot \cos(\frac{2\pi n}{N}) + 0.14128 \cdot \cos(\frac{4\pi n}{N}) - 0.1168 \cdot \cos(\frac{6\pi n}{N})$ with $0 \leq n \leq N - 1$

Table 3. List of window lengths and hop sizes

Configuration	**Window length**	**Hop size**
Short	2,028	1,014
Medium	22,050	11,025
Long	44,100	22,050

K-Nearest Neighbors (KNN) Since each windowing configuration produces varying amounts of feature vectors per data point (shorter windows produce more feature vectors), values of k that linearly scales with the amount of feature vectors per data point were incorporated. When a window length is long, a small number of feature vectors are produced. Inversely, when the window length is short, the amount of feature vectors produced will be higher.

Random Forest Through preliminary testing with the IRMAS dataset, it is observed that a small number of estimators can be detrimental to the performance of the classifier. Consequently, a wide set of values are tested during hyperparameter optimization to search for the optimal performance in each experiment.

Support Vector Machines We employed a stochastic gradient descent (SGD) linear classifier (using the hinge loss function) in combination with the Nyström kernel approximation method. The Nyström kernel approximation method also allows for the modification of the the number of components used for representation, which determines the size of the explicit feature map. Using the kernel trick, particularly on large datasets can entail large amounts of computation, as computations of the kernel function can be costly. With a larger feature map, a closer approximation of the kernel is achieved. Hence, a wide set of values for both the alpha value of the SGD classifier, as well as number of components of the Nyström kernel approximator were explored. We employed the "One-vs-all" approach to classify the multiclass dataset.

Dataset We used the Instrument Recognition in Musical Audio Signal (IRMAS) dataset. This dataset consists of 9,579 files of 16-bit stereo audio, sampled at 44.1kHz. Containing audio excerpts from 11 classes of pitched instruments, real world music from a variety of different eras and genres were incorporated in this dataset.

4 Methodology

Feature selection We employed the ANOVA-based feature selection method. Like other filter methods, it ranks individual features before choosing a feature, and iteratively adds a feature to the current feature set (Jović, Brkić, & Bogunović, 2015). The ANOVA-based feature selection method works by producing the p-value associated with the distribution of the features and the labels. Features are then ranked and selected based on its correlation to the output variable (or label).

Normalization Prior to feeding data to the respective models, features are scaled to a range within 0 and 1, using min-max normalization (J. Han, Kamber, & Pei, 2011).

Aggregation strategy Since each audio excerpt is extracted into multiple feature vectors, an aggregation strategy is employed to aggregate multiple feature vectors back into a single data point. We employed the late fusion strategy (also known as decision level fusion), a strategy which was also adopted by Slizovskaia et al. (Slizovskaia et al., 2016). In particular, we combined multiple output labels via majority voting. To produce the final label from a set of labels obtained, a vote is assigned for each instance of the label in the set. Subsequently, the final label is derived from the label which has the most votes. This can be done simply using a mode function, which returns the most frequently occurring number from a set of numbers.

5 Results and analysis

We report the results of our first set of experiments in table 4.

Table 4. Results of the first set of experiments. The best accuracy per window function are presented in bold, and the best overall accuracy is underlined.

Window configuration			Classification accuracy		
Function	Length	Hop size	SVM	KNN	RF
Hann	2,028	1,014	0.441	0.437	**0.451**
	22,050	11,025	0.417	0.389	0.422
	44,100	22,050	0.428	0.422	0.410
Hamming	2,028	1,014	0.444	0.442	**0.455**
	22,050	11,025	0.419	0.423	0.423
	44,100	22,050	0.434	0.412	0.408
Blackman-Harris	2,028	1,014	0.437	0.424	**0.447**
	22,050	11,025	0.410	0.420	0.418
	44,100	22,050	0.424	0.406	0.403

5.1 Window configuration

The first point of interest in our analysis is the configuration (function and length) of the windows used for feature representation in our classification pipeline.

Window function We can observe in table 4 that the accuracies when comparing the same configuration for different window functions look very similar. A T-test confirms that there is no statistically significant difference ($p > 0.05$) and therefore we cannot draw any robust conclusion based on our current data. However, the Hamming window tends to performs best in the majority of experiments, followed by the Hann window and the Blackman-Harris window in the last position. This trend needs to be confirmed in further experiments.

Window length We observe from table 4 that the performance of the models seems to improve with a shorter window length. That improvement was found to be statistically significant ($p < 0.05$) using a paired T-test, confirming a short window to be optimal. This suggests that acoustic events of interest which capture differences between musical instruments tend to be shorter than 500ms.

5.2 Audio Features

Our second point of interest is the optimal set of features required for feature representation. Table 5 summarizes the relative frequency of each feature in the best performing model for each classifier, and averaged over classifiers.

Table 5. Results of the feature selection experiment. MFCC and BFCC are condensed for lack of space.

Window configuration		Feature frequency			
Domain	Feature	SVM	KNN	RF	Overall
Temporal	Log attack time	0%	66%	11%	25%
	Zero crossing rate	100%	100%	100%	100%
Spectral	Spectral flatness	100%	100%	100%	100%
	Spectral roll-off	100%	100%	100%	100%
	Spectral flux	100%	100%	100%	100%
	Spectral centroid	100%	100%	100%	100%
	Spectral variance	100%	100%	100%	100%
	Spectral skewness	100%	100%	100%	100%
	Spectral kurtosis	100%	100%	100%	100%
Cepstral	MFCC (13 features)	95%	100%	100%	98%
	BFCC (13 features)	89%	98%	97%	94%

The ANOVA based feature selection process iteratively adds new features to the existing feature set until it has exhausted all the features available. In each iteration, the features are used as an input to a machine learning classifier, subsequently producing a set of performance metrics that are used to evaluate the performance of the feature set. Each experimental configuration produced 38 sets of scores, each corresponding to the performance of the model at each iteration of the feature selection process.

In order to answer the third research question, best performing feature sets from the experiments were analyzed. A feature set is deemed better than the other if it produces a better performing model, given the same experimental configuration. In total 27 best performing feature sets (9 per classifier) were obtained, each corresponding to different experimental configurations and classifiers.

All of the features drawn from the spectral domain saw full usage by the best performing models across all three classifiers. From the cepstral domain, the first 11 coefficients of both MFCC and BFCC feature sets were fully employed by all of the experiments. Lastly, from the temporal domain, the zero-crossing rate saw usage from all experiments, while the log attack time was only sparingly used by the KNN classifier. Overall, only 33 out of 38 features are common in usage across all of the experiments conducted. This common feature set consists of 1 out of 2 features from the temporal domain (zero crossing rate), all features from the spectral domain, and 22 out of 26 features from the cepstral domain (all but MFCC and BFCC coefficients 12 and 13). The random forest classifier, which performed best out of other classifiers required 37 features to produce its best performing model, while SVM and KNN required 34 and 37 features respectively.

MFCC features were overall judged to be useful, which supports the conclusion made by Eronen on the effectiveness of MFCC in musical instrument recognition (Eronen, 2001). On the other hand, BFCC features comparatively saw less usage.

5.3 Classifiers

In order to answer this research question, we made comparisons of the performances of the three classifiers obtained from the experiments. As seen from Table 4, the best overall performance was achieved by the random forest classifier, using the Hamming window and the short window length, but it was not found to be statistically significant using a T-test and therefore we cannot draw any robust conclusion based on our data.

6 Conclusion

In this paper we provided an investigation into three aspects of instrument classification and their impact on performance: window function configuration, audio features, and classification algorithm. We use the results of our experiments in order to answer the three research questions formulated in the introduction. We found the Hamming window to be the most effective window function out of the three window functions explored. The best windowing configuration found is the Hamming window in conjunction with the short window length (**RQ1**). We found features from the spectral domain to lead to the best performance. The majority of features from the cepstral domain were also found in the best performing feature sets, with the MFCC seeing more usage than the BFCC. From the temporal domain, the log-attack time feature was shown to be the least-used feature in this paper, while the zero-crossing rate feature saw full usage by all experiments (**RQ2**). Finally, we found that the random forest classifier outperformed other classifiers, but the results were not found to be statistically significant (**RQ3**).

References

Alías, F., Socoró, J. C., & Sevillano, X. (2016). A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds. *Applied Sciences*, *6*(5), 143.

Benetos, E., Kotti, M., & Kotropoulos, C. (2006). Musical instrument classification using non-negative matrix factorization algorithms and subset feature selection. In *2006 ieee international conference on acoustics speech and signal processing proceedings* (Vol. 5, pp. V–V).

Bosch, J. J., Janer, J., Fuhrmann, F., & Herrera, P. (2012). A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In *Ismir* (pp. 559–564).

Brent, W. (2009). Cepstral analysis tools for percussive timbre identification. In *Proceedings of the 3rd international pure data convention, sao paulo, brazil.*

Eronen, A. (2001). Comparison of features for musical instrument recognition. In *Proceedings of the 2001 ieee workshop on the applications of signal processing to audio and acoustics (cat. no. 01th8575)* (pp. 19–22).

Eronen, A., & Klapuri, A. (2000). Musical instrument recognition using cepstral coefficients and temporal features. In *2000 ieee international conference on acoustics, speech, and signal processing. proceedings (cat. no. 00ch37100)* (Vol. 2, pp. II753–II756).

Fuhrmann, F., Haro, M., & Herrera, P. (2009). Scalability, generality and temporal aspects in automatic recognition of predominant musical instruments in polyphonic music. In *Ismir* (pp. 321–326).

Fuhrmann, F., et al. (2012). *Automatic musical instrument recognition from polyphonic music audio signals* (Unpublished doctoral dissertation). Universitat Pompeu Fabra.

Gulzar, T., Singh, A., & Sharma, S. (2014). Comparative analysis of lpcc, mfcc and bfcc for the recognition of hindi words using artificial neural networks. *International Journal of Computer Applications*, *101*(12), 22–27.

Gururani, S., Sharma, M., & Lerch, A. (2019). An attention mechanism for musical instrument recognition. *arXiv preprint arXiv:1907.04294*.

Han, J., Kamber, M., & Pei, J. (2011). Data transformation and data discretization. *Data Mining-Concepts and Techniques, ed: Morgan Kaufmann*, 111–112.

Han, Y., Kim, J., & Lee, K. (2016). Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(1), 208–221.

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

Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (mipro)* (pp. 1200–1205).

Lostanlen, V., & Cella, C.-E. (2016). Deep convolutional networks on the pitch spiral for musical instrument recognition. *arXiv preprint arXiv:1605.06644*.

Lyon, R. F. (2010). Machine hearing: An emerging field [exploratory dsp]. *IEEE signal processing magazine*, *27*(5), 131–139.

Park, T., & Lee, T. (2015). Musical instrument sound classification with deep convolutional neural network using feature fusion approach. *arXiv preprint arXiv:1512.07370*.

Patil, K., Pressnitzer, D., Shamma, S., & Elhilali, M. (2012). Music in our ears: the biological bases of musical timbre perception. *PLoS Comput Biol*, *8*(11), e1002759.

Seipel, F. (2018). Music instrument identification using convolutional neural networks.

Slizovskaia, O., Gómez Gutiérrez, E., & Haro Ortega, G. (2016). Automatic musical instrument recognition in audiovisual recordings by combining image and audio classification strategies. In *Großmann r, hajdu g, editors. proceedings smc 2016. 13th sound and music computing conference; 2016 aug 31; hamburg, germany. hamburg (germany): Zm4, hochschule für musik und theater hamburg; 2016. p. 442-7.*