Classification of Percussive Sounds Using Wavelet-Based Features

Michelle Daniels

ECE 251C
Fall 2010

1 Introduction

Automatic classification of percussive sounds is a machine listening task in which digital audio recordings of sounds produced by various percussion instruments are identified by a computer. This kind of classification task has a broad range of applications, including automatic drum transcription (in which it is necessary to know exactly what instrument is playing and when), genre identification (in which the instrumentation of a song might provide clues about its genre), or real-time interactive electro-acoustic music performance (in which a computer makes decisions about what material to play based on the real-time analysis of sounds generated by a performer on an acoustic instrument).

The classification process usually begins with a training stage, in which a classifier is trained on certain features extracted from known, labeled recordings. Depending on the number of training samples available and the type of classifier used, training can be quite time-consuming and is generally performed off-line. Once the classifier has been trained, however, classification of new samples can potentially be performed in real time. Audio classifiers generally use either temporal or spectral features, or some combination of the two. Because of the transient nature of percussive sounds and the wide variety of timbres produced by percussion instruments, techniques used for the classification of other instruments or sounds are often not immediately applicable to percussion, resulting in a body of work specifically focused on this family of sounds. In such work, classifiers for percussive sounds have traditionally used temporal features, which could include zero crossing rate and temporal centroid, and/or spectral features, which might include MFCCs and spectral centroid [4][12][3].

As an alternative to these more traditional features, this project explored the use of wavelet-based features for the classification of percussive sounds. First, a variety of percussive sounds were analyzed using the Discrete Wavelet Transform in an attempt to identify wavelet-based features that effectively capture the differences between various percussion instruments. Once identified, these features were then used to automatically distinguish between different percussion instruments in a classification task. Experiments were performed comparing the success rate of classification using wavelet-based features to the success rate of classification using the commonly used Mel-Frequency Cepstral Coefficients (MFCCs), and the wavelet-based features were found perform comparably to MFCCs.

2 Audio Features

For machine listening tasks, features are any measurable aspect of an audio signal which might distinguish one class of signals from another for the purposes of classification, clustering, or other applications. The ”best” features are those which produce the most accurate results for a task with the least computational expense. Clearly, smaller feature sets are preferred because a lower-dimensional feature vector can be more efficiently classified than one with many components. Smaller feature sets can also help to minimize the risk of overfitting a model during the training stage. The choice of features is therefore complicated, and often the ideal set of features is specific to a particular class of signals. For percussive sounds, because of their transient nature characterized by sharp attacks and minimal sustain, the ability of wavelets to provide both time and frequency localization is appealing as a source of features. In this project, wavelet-based features were compared with Mel-Frequency Cepstral Coefficients (MFCCs).

2.1 MFCCs

Mel-Frequency Cepstral Coefficients originated in speech processing [CITATION HERE??], where they have been used in speech recognition applications for over 30 years [8]. Their use in general audio and music similarity-based applications is a more recent development [7][1].

2.1.1 The Mel Scale

A Fourier transform produces a spectrum with bins linearly spaced in frequency. However, human perception of frequency (pitch perception) is not linear. Instead, the perceived difference in pitch between two low frequency signals is smaller than the perceived difference in pitch between two higher frequency signals, even if the low and high frequency pairs are spaced the same number of Hertz apart. The Mel frequency scale was developed to represent this non-linear perception of frequency. By converting a spectrum on a linear frequency scale to the Mel scale, lower frequencies are given more weight than higher frequencies in an attempt to provide a signal analysis that is more consistent with human perception. As with a linear frequency spectrum, different numbers of frequency bands can be used to give the desired frequency resolution. The MIRtoolbox uses 40 bands by default, resulting in a significantly lower frequency resolution, especially at higher frequencies, than the original Fourier spectrum. This is the number of bands used at the standard CD sampling rate of 44.1kHz. The recordings used for this project, however, were sampled at 96kHz, so the default number of bands used for the Mel-scale spectrum was not necessarily appropriate in this case. I chose to use 60 bands instead, but ideally this number would be experimentally determined.

2.1.2 Computation

MFCCs are cepstral coefficients on a Mel frequency scale rather than the traditional linear frequency scale. They are computed by first windowing a signal and computing the spectrum using the DFT. The log-magnitude of the spectrum is computed and then converted to the Mel frequency scale, a process typically performed using a filter bank of overlapping triangular bandpass filters whose bandwidth increases as their center frequency frequency increases. After conversion to the Mel scale, a discrete cosine transform (DCT) is performed as a decorrelation step to obtain the final coefficients.

2.1.3 MFCCs as features

The lowest coefficient represents the overall signal energy and is therefore typically ignored for classification tasks, especially those in which signals from the same class might have different dynamic ranges. Other coefficients give information about the spectral envelope of the signal, and in some applications the change in coefficient values between consecutive analysis frames has been used to give some measure of temporal variation in the spectrum.

One decision that must be made when applying MFCCs is how many coefficients to use. Typically, only a subset of coefficients is used. In the MIRtoolbox, the default setting uses coefficients 1:13, with coefficient 0 ignored. The use of thirteen coefficients is apparently standard in speech recognition applications, but as few as five coefficients have been used successfully as features for other tasks such as genre recognition [10]. As mentioned above, the recordings used in this project were sampled at more than twice the typical audio sampling rate for MIR applications. Therefore, the default use of 13 coefficients now representing more than twice the frequency range used in other applications was not necessarily an appropriate choice.

In order to determine the ideal number of coefficients to use in classification, I performed classification tests using varying number of coefficients from 5 to 26. I expected to find some kind of trend with a clear optimal number of coefficients, but instead, after 10 coefficients were reached, there was no obvious trend to the data. The results are shown in Figure 1. The classification success rate varied from 53% to 68%, with the highest success rates occuring with 18 or 19 coefficients. As a result, I chose to use 18 coefficients for the remainder of my experiments.

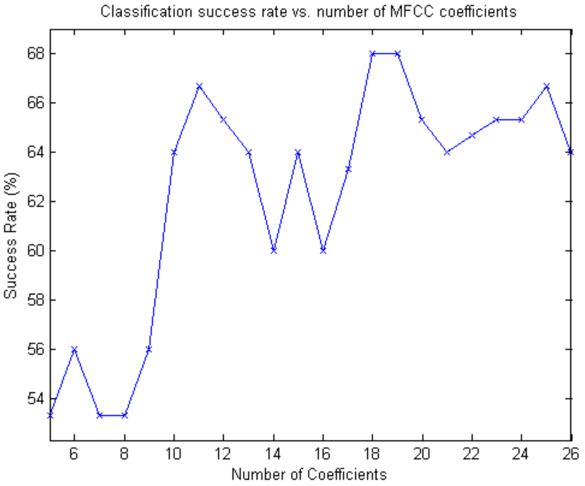

Figure 1: Classification Success vs. Number of MFCC Coefficients

2.2 Wavelets

2.2.1 Prior Work

Wavelets have occasionally been used in the past as part of automatic music classification systems, but this area has either not been well explored or not well documented. For example, one paper [9] mentions using wavelets for classification of percussive sounds, but only as one small part of a large feature set containing multiple temporal and spatial features, and there is no discussion about whether using wavelets introduced any benefit over the use of the other features alone. Another paper [10] uses wavelets to derive rhythmic information about a musical recording for genre classification, not for identification of specific instruments. A third paper [6] utilizes wavelet sub-band power, derived using the 8-coefficient db4 wavelet, as a feature for general sound classification and additionally uses a wavelet-based pitch detection algorithm to determine a pitch feature, but the source audio is all down-sampled to 8kHz, meaning that percussive sounds, which have significant energy in higher frequencies, are not well-represented. The paper that is perhaps the most relevant [11] does examine the performance of their classifier using wavelets vs. MFCCs vs. Short-Time Fourier Transform features, noting that the wavelet approach performs comparably, but it is also not focused on percussive sounds in particular and doesn't include a very detailed description of the work. In that work the 4-coefficient ”DAUB4” (db2) wavelet is used. Many of these papers either do not mention the type of wavelet used, or mention it only in passing with no explanation of the choice of wavelet. Therefore, there appears to be significant room for further exploration of this area.

Finally, given a set of features, there are a variety of machine learning techniques that can be applied to perform the actual classification. Past work on classification of percussive sounds has utilized techniques including K-Nearest-Neighbor, Kernel Density Estimation [4], and Support Vector Machines [12][6], or unsupervised classification algorithms such as Agglomerative Clustering [3], but this project is not intended to produce a comparison between different classifiers, only between different features. Therefore, for simplicity, all classification experiments in the project will be performed using a K-Nearest-Neighbor algorithm implemented in the software package Weka [13].

2.3 Wavelet-Based Features

2.3.1 Choice of Wavelet

The ideal choice of wavelet would be the one that best provides meaningful distinguishing features while having a short enough filter length to be computationally practical for real-time use and maintain a reasonable amount of time localization after repeated filtering. I performed some classification experiments using the Daubechies 1 through 8, Symlet 2 through 8, and Coiflet 1 through 5 wavelets to get a broad sense of which wavelet provided the best results. As shown in Figure 2, in the first classification task, which used separate training and test data, all of the wavelets tested performed comparably. I suspect that the differences were not statistically significant between most of the wavelets. However, the db4, db5, and sym5 wavelets (scoring identically) were slightly superior to the rest, while db1 performed the worst. In general, since the results using the Daubechies family of wavelets seemed to be representative of the results with the other wavelet families (and in general slightly better), the rest of the project focused solely on comparisons within the Daubechies family.

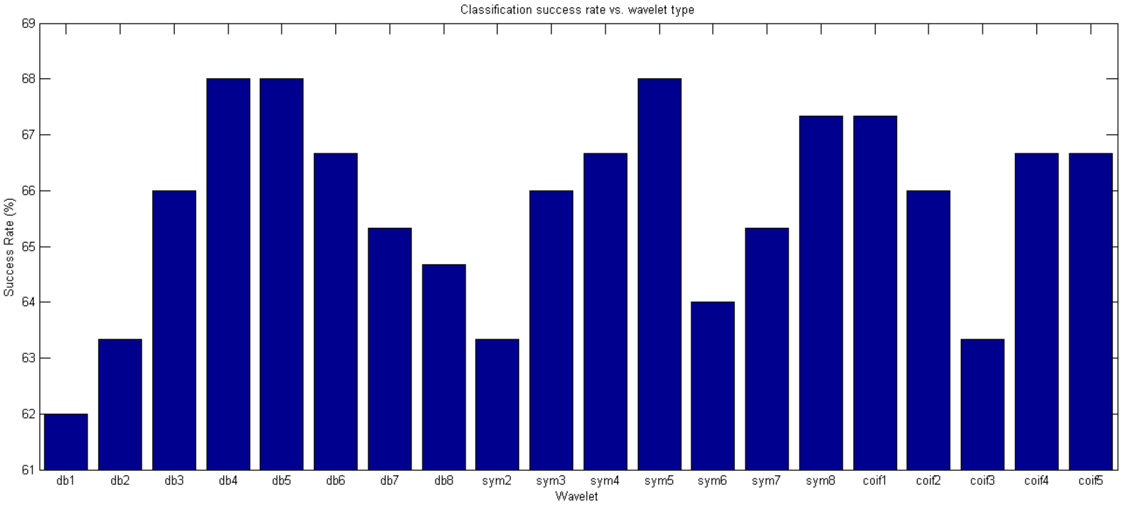

Figure 2: Classification Success vs. Wavelet Type

In the experiment described above, three sets of wavelet-based features were used: the ratio of mean energy between adjacent subbands, the temporal centroid of each subband, and the difference between temporal centroids in adjacent bands. Performing the same experiment (on the Daubechies family only) but using subsets of these three features reveals interesting results that suggest that the choice of optimal wavelet varies depending on the features used. These results are shown in Figures 3 and 4. Notice that for the first pair of features, the db5 wavelet is clearly optimal, whereas for the second pair of features, the db6 wavelet is optimal but has more competition from other wavelets. This suggests the possibility that the optimal choice may in fact be to use different wavelets for each feature.

2.4 Choice of Wavelet Decomposition Level

Another parameter relating to the choice of wavelet is the choice of how many levels of decomposition should be performed before extracting features. Because the signals I was analyzing were all 4096 samples in duration, the maximum number of times I could downsample was 12, so I initially chose 12 levels for my decomposition. With the Haar wavelet this results in only one coefficient in the

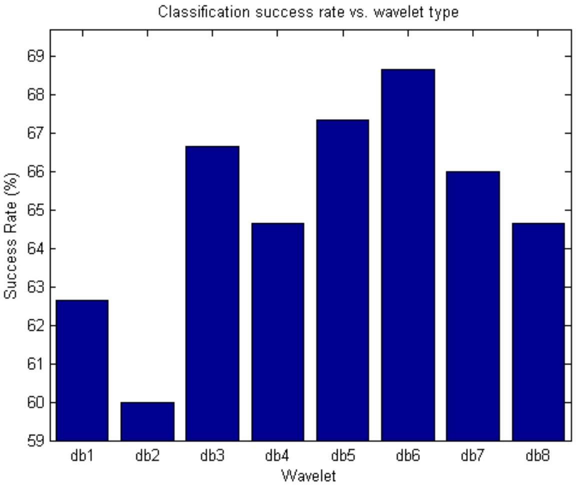

Figure 3: Classification Success vs. Wavelet Type (Using Subband Mean Ratios and Temporal Centroids)

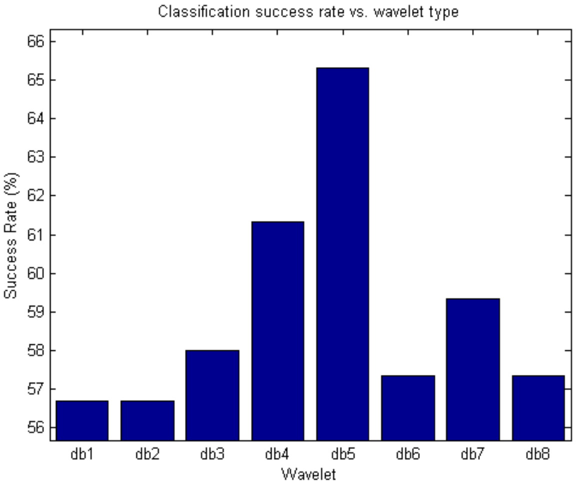

Figure 4: Classification Success vs. Wavelet Type (Using Subband Mean Ratios and Temporal Centroid Differences)

approximation and lowest detail level, so I suspected that this was overkill. To test this hypothesis, I ran classification experiments using the Daubechies wavelet family at various decomposition levels for feature extraction. The resulting success rate at each of 8 different levels, averaged across db1 through db8 wavelets, is shown in Figure 5. Unlike the MFCCs, where the success rate was not a smooth function of the number of coefficients, with the wavelet decomposition there is a reasonably smooth increase in success rate as the number decomposition levels increases, up until the highest number of levels, where as expected, the small size of the resulting final band was likely more misleading than helpful. While the highest average success rate occurred at 11 levels, the highest rate for an individual wavelet was the db6 wavelet at 8 levels (71.33%). The db4 wavelet at 8 levels was almost as effective as db6, with 68.67% success. The highest success rate at 11 levels was 69.33%, so even though the average success rate was higher elsewhere, I chose the db4 wavelet with an 8-level decomposition as the ideal balance of computational complexity (filter length and decomposition levels) and success rate. Later comparisons with MFCCs were all performed using this wavelet.

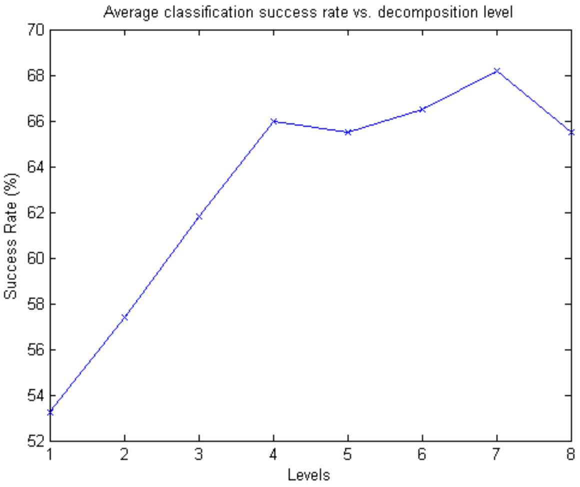

Figure 5: Average Classification Success vs. Decomposition Level

2.4.1 From Tzanetakis Paper

Subband means and Subband STDs were tried, but were found to have little effect on classification accuracy.

2.4.2 Subband Centroid

Gives a measure of the distribution of energy over time in each frequency subband. (normalized) Would depend on training and test samples being properly aligned - one option, ratios, or maybe diff from first subband's centroid to current one?? Note: normalized to 0,1

3 Experiments

3.1 Audio Material

All analysis as well as the classification experiments were performed using samples of percussion instruments recorded for use in William Brent's PhD thesis on the timbre of percussive sounds [2].

Recordings at 96kHz/24-bits were made in a studio setting using professional-quality microphones and consisting of percussive sounds generated by a percussionist striking various instruments at five different dynamic levels using a variety of mallets. All recordings used in this project were recorded in the same location with the same microphone, eliminating one possible source of bias in classification. Although the recordings included the full sustain and decay of each instrument strike, for practical reasons the scope of this project was limited to the attacks of percussive sounds only, so the recordings were segmented into 4096-sample excerpts in which the attack of a percussive sound occurred around sample 1024. This limitation is not overly restrictive, given that in practice it may be desirable to perform classification in real time with minimal latency for electro-acoustic musical performance. In such a case, it would be necessary to identify the source instrument from as few samples as possible. The placement of each attack at sample 1024 was chosen such that as much of the initial sustain and decay of the instrument as possible could be seen without placing the attack at the very beginning of an excerpt where windowing for the Fourier domain for certain kinds of analysis would nearly eliminate the attack. In a real-world implementation of this system, provisions would have to be made for input signals at varying sample rates, and some kind of onset detection might have to be performed to identify the start of an attack. The time-alignment of the attack in the analysis frame would need to be considered as well, especially when comparing two attacks that started at different instants. These practical problems have been avoided for the limited scope of this project, but were kept in mind when considering the relative merits of different audio features for classification.

3.2 Tools and Implementation

Audio recordings were segmented using the Cool Edit Pro audio editing software. All analysis was then performed in MATLAB using the MATLAB Wavelet Toolbox to visualize and extract wavelet-based features and using the free MIRtoolbox [5] to compute MFCCs. The MIRtoolbox was chosen because of its popularity among the music information retrieval (MIR) community, so that the MFCC features were generated in a way more fairly compared to other research than a custom implementation could have been. After the wavelet and MFCC features were generated in MATLAB, they were written to Attribute-Relation File Format (ARFF) data files for use in the popular and freely available Weka machine learning software [13]. This software includes a GUI-based tool for experimenting with machine learning algorithms, and this was used for all classification experiments, partly for simplicity and partly (as with the MFCC computation) to avoid biasing results by using a custom implementation.

3.3 Classifier

3.4 Classification Experiments

All classification was performed using separate training and testing data. The training data set contained 5 samples of each instrument/mallet combination, and the testing data set generally also contained 5 instances of each combination, except in a few cases where 10 instances were available and some in which no testing instances were available at the time. The classifier used was the ”NNge” classifier in Weka, which is similar to a nearest-neighbor algorithm. This was classifier was chosen because it worked reasonably well out of the box with no parameter tweaking required. However, other choices of classifiers may be more appropriate for this kind of task, but that is outside the scope of this project. Instead, I have used the same classifier in all experiments and focused instead on identifying an effective feature set.

3.4.1 Classification by Instrument and Mallet

3.4.2 Classification by Instrument Only

4 Results

Describe instruments where MFCCs or Wavelets are superior and explain why.

Describe results using different wavelets.

Describe results using different levels of decomposition.

Interesting that the MFCCs do a bad job of distinguishing tom from snare with no snares...

5 Conclusions and Future Work

Mention that it's difficult to draw broad conclusions with limited size data set, so many possibilities for MFCC parameters, other spectral features, etc., but that wavelets seem to work at least as well as MFCCs without extensive parameter tweaking, etc.

Mention exploring wavelet tree, that there may be interesting information in higher frequencies.

Possibly use different wavelets for each feature

References

[1] Adam Berenzweig, Beth Logan, Daniel P.W. Ellis, and Brian Whitman. A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures. *Computer Music Journal*, 28(2):63–76, June 2004.

[2] William Brent. *Physical and Perceptual Aspects of Percussive Timbre*. Phd dissertation, UCSD, 2010.

[3] Fabien Gouyon, Francois Pachet, and Olivier Delerue. On the Use of Zero-Crossing Rate for an Application of Classification of Percussive Sounds. In *Proc. of the COST G-6 Conference on Digital Audio Effects (DAFX-00)*, Verona, Italy, December 2000.

[4] P. Herrera, A. Dehamel, and F. Gouyon. Automatic labeling of unpitched percussion sounds. In *Proc. of the 114th AES Convention*, Amsterdam, The Netherlands, March 2003. Audio Engineering Society.

[5] Olivier Lartillot and Petri Toiviainen. A Matlab Toolbox for Musical Feature Extraction From Audio. In *Proc. of the 10th International Conference on Digital Audio Effects, DAFx-07*, Bordeaux, France, September 2007.

[6] Chien-chang Lin, Shi-huang Chen, Trieu-kien Truong, and Yukon Chang. Audio classification and categorization based on wavelets and support vector Machine. *IEEE Transactions on Speech and Audio Processing*, 13(5):644–651, September 2005.

[7] Beth Logan. Mel Frequency Cepstral Coefficients for Music Modeling. In *Proc. International Symposium on Music Information Retrieval*, 2000.

[8] Paul Mermelstein. Distance Measures for Speech Recognition – Psychological and Instrumental. In *Proc. Joint Workshop on Pattern Recognition and Artificial Intelligence*, Hyannis, MA, June 1976.

[9] Adam Tindale, Ajay Kapur, and George Tzanetakis. Retrieval of Percussion Gestures Using Timbre Classification Techniques. In *Proc. International Conference on Music Information Retrieval*, 2004.

[10] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

[11] George Tzanetakis, Georg Essl, and Perry Cook. Audio Analysis using the Discrete Wavelet Transform. In *Proc. WSES Int. Conf. Acoustics and Music: Theory and Applications (AMTA 2001)*, Skiathos, Greece, 2001.

[12] D. Van Steelant, K. Tanghe, S. Degroeve, B. De Baets, M. Leman, and J.-P. Martens. Classification of Percussive Sounds Using Support Vector Machines. In *Proc. of the Annual Machine Learning Conference of Belgium and The Netherlands*, Brussels, Belgium, 2004.

[13] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, San Francisco, 2nd edition, 2005.