HMM Analysis and Synthesis of Acoustic Drum Signals

Nicholas J. Bryan
Center for Computer Research in Music and Acoustics

***Abstract*— Hidden-Markov Models (HMMs) have been widely used for speech processing, understanding, and synthesis with great success. The purpose of this work is to apply this prior knowledge and investigate the effectiveness of HMMs on short-duration percussive musical signals. Three main topics of interest are investigated: isolated instrument recognition, isolated rhythm transcription for the purpose of genre recognition, and isolated instrument synthesis. Overall, satisfactory results were achieved with clear motivation for improvement.**

I. Introduction

PERCUSSION, drums, and other rhythmic acoustic signals and patterns are an integral aspect of modern day music. Entire music genres, careers, and extensive software applications are based off of rhythm or musical patterns through time, providing a large motivation to learn and model such information. More specifically, large databases of previously recorded drum sounds are common place throughout the music industry with little or no method of automatically labeling, identifying, or searching with respect to musical parameters, forcing manual searching with real-time auditory assessment. To attack this issue and provide insight on such problems as "search-by-rhythm" or "search-by-rhythm-genre", isolated instrument recognition, short-duration rhythmic recognition, and isolated re-synthesis of acoustic drum set signals of typical performance are investigated. To model the time-series information of both the acoustic pressure information of the musical samples as well as the rhythmic information of a musical measure, continuous and discrete observation Hidden Markov models are used respectively. Once the isolated instrument recognition HMM models are complete, synthesis can be performed using the learned HMM models. For an overview of percussion transcription techniques see [1].

II. Feature Vectors

With respect to the input feature vectors, the input time-domain audio signals are converted to Mel-frequency cepstral coefficients (MFCCs) [2]. Introduced in [3], MFCCs attempt to more closely model the human auditory response, while exploiting the decorrelating property of the cepstrum [4]. The cepstrum of an audio signal can informally be defined as the inverse Fourier transform of the logarithm magnitude of the Fourier transform. The synthesis step (inverse transform) in application actually uses the discrete cosine transform and in [5] was shown to be effective in approximating principal component analysis. Thirteen MFCC coefficients are generated out of a 512-point FFT using .025 seconds windows overlapping every .010 seconds. Ideally, the MFCC data effectively captures a pitch-independent frequency response of the audio signal over time. Using such feature vectors also allows for a respectable synthesis of the instrument sounds.

III. Isolated Recognition

To break down the problem of isolated acoustic drum set recognition, six basis classes are used to represent each major sub-instrument of a typical drum set (s = snare drum, b = bass drum, h = hi-hat, t = toms, c = cymbals, and si = silence). Two-hundred audio samples for each basis class are used for learning from a commercially available drum sample database [6]. The EM (or in this context the Baum-Welch) algorithm is used for learning the continuous observations using a single Gaussian mixture model [7], [8]. To effectively model the remaining combinations of drum sounds such as a snare drum and bass drum played simultaneously (sb = snare + bass), combination data is created via random sampling, adding together, and amplitude normalization from the basis class audio files [9]. Using all physically realizable combinations (no more than four simultaneous sub-instruments at a time as well as silence exclusion) of the six basis classes, a total of 28 overall classes were used. Additionally, the level of combination of each class was identified with a corresponding complexity level (basis class = 1, two added together = 2, etc). For each class, a 5-state left-to-right state-sequence HMM model was generated. For recognition, a maximum log-likelihood classification is used.

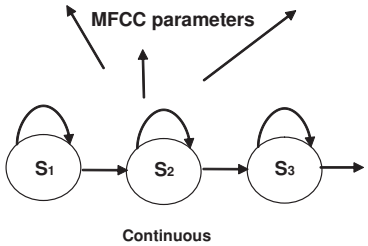

Fig. 1. Continuous Observation HMM State Sequence

IV. Rhythmic Transcription for Genre Recognition

Once each isolated recognition model is learned, musical sequence or rhythmic recognition of audio files can be applied

by using multiple instances of isolated recognition. The sequence or rhythmic recognition must decode each event of the sub-instruments or sub-instrument combinations. Specifically, isolated recognition must be processed on each event within the rhythmic drum pattern (typically one to two measures), where an event occurs at every smallest division of musical time or beat. Additionally, each rhythm data example must be normalized with respect to time. Once time normalized, the sequence recognition becomes straight forward and independent of beat detection errors. Reason 3.0 with Dr. REX loop player (a commercially available music production software) was used to set a standard 120 beats per minute for each multi-measure example of a standard 4/4 time signature with 16th note quantization (i.e. for a two measure pattern, 32 isolated recognition classifications well be made). For training and

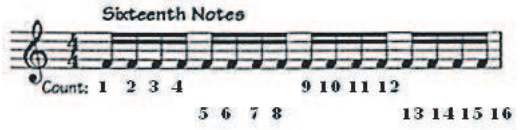

Fig. 2. Isolated Recognition for Every 16th Note Division

testing purposes, a small collection of fifty rock rhythmic patterns and fifty hip-hop patterns (previously defined by genre) were decoded into discrete events consisting of the isolated recognition classes. Once the decoded instrument patterns are generated, the data can then be used to model the rhythmic characteristics of the respective genre. A discrete-observation (28 class) HMM model can then be used to classify using a maximum log-likelihood approach. The two-class dataset illustrated significant results considering the minimal number of examples. See Results

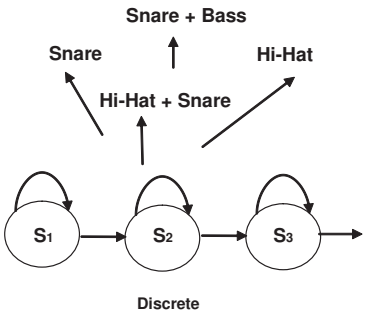

Fig. 3. Discrete Observation HMM State Sequence

V. Synthesis

Independent of the rhythmic transcription, the isolated basis class HMM models can be used to re-synthesize audio wave forms. The trained HMMs and an optimal state sequence can generate the modeled MFCC parameters using the expected value observations for each state. Unfortunately, while the MFCC features give an approximate form of the frequency transfer function of the audio signals, difficulty arises when attempting to invert the MFCC process (all phase information of the signal is lost by taking the logarithm of the magnitude). As a result, a source-filter model can be used with a shaped-noise input signal to excite the synthesized parameters [10]. Pink noise (filtered white noise) is used as the excitation and convolved with the frequency response parameters (MFCCs) for the final output audio signal. Pink noise can be defined by a power spectral density proportional to the reciprocal of the frequency. The overall synthesis model can be illustrated in fig. 4, similar to [11]. Each time window of MFCC data essentially acts as a dynamic filter, shaping the spectrally flat (or sloped) noise. Ideally, the generated state sequence can be used to parametrically control the synthesis of the audio waveform such as controlling the attack, decay, sustain, and release parameters of the drum signals. The parameters of the synthesis were unfortunately difficult to control with respect to auditory evaluation due to the transient behavior or drum signals (this is not the case for the typical speech application). Moreover, numerous HMM models of varying state sequence lengths were used with little noticeable improvement. Overall, a 5-state sequence was used.

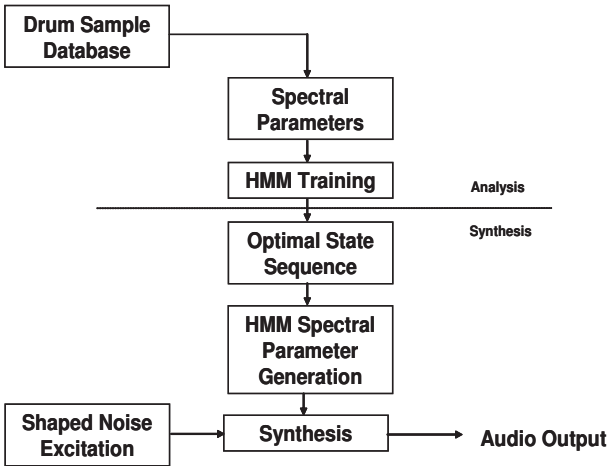

Fig. 4. Synthesis Model

VI. Results

Overall results proved moderately successful. The following will be a presentation and discussion of the results for the isolated recognition, rhythmic transcription for genre recognition, and synthesis. For all confusion matrices, the rows represent the known correct classification, while the columns represent the predicted classification.

A. Isolated Recognition

With respect to isolated recognition, two main training and testing schemes were implemented. Initially, 10-fold cross-validation was used on the six basis class (200 examples/class) HMM models only. Very accurate results were obtained. See the confusion matrix in fig. 5. After basis class verification, 10-fold cross-validation was used to train/test all 28 classes together and can be seen in fig. 13. Unfortunately, classification accuracy greatly decreases as the complexity level increases, making musicological analysis of the rhythm transcription less useful. The misclassifications, however, are typically educated

Confusion Matrix (%)

	s	b	c	h	t	si
s	100	0	0	0	0	0
b	0	99	0	0	1	0
c	0	0	97.6	0	0	2.38
h	5.5	0	0	94.5	0	0
t	0	0	0	0	99.5	0.5
si	0	0	0	0	0	100

Fig. 5. Basis Class Confusion Matrix (%)

in some manner (i.e. a snare drum gets misclassified as a snare + bass drum) and can be interpreted as resonances of the basis classification. A general analysis of the complexity vs. classification accuracy result can be seen in fig. 9. See fig. 13 for a confusion matrix of all classification.

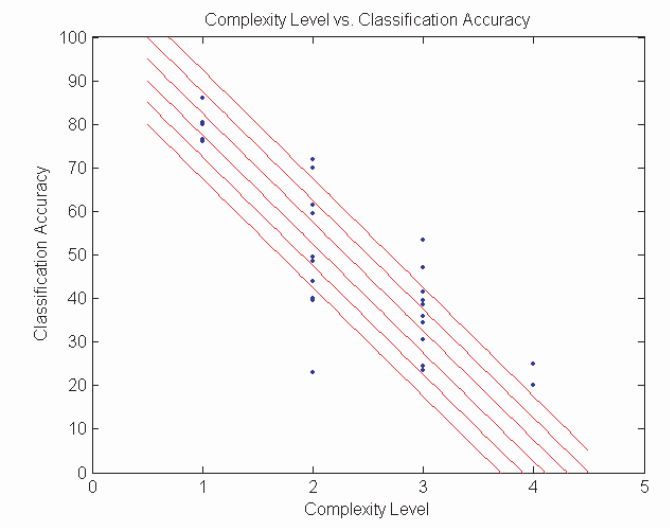

Fig. 6. Complexity vs. Classification Accuracy

B. Rhythmic Transcription for Genre Recognition

Although the performance of the isolated recognition significantly decreases with complex instrument combinations, accurate genre recognition using sequential isolated recognition was obtained. Using leave-one-out cross-validation and the minimal database of fifty hip-hop and fifty rock classified two-measure audio files, genre-classification by rhythmic transcription provided around 76-80% classification accuracy. See the confusion matrices for multiple tests using varying state-sequence lengths. It should be noted that a large improvement on the classification accuracy should result with a significantly larger database, providing promise for rhythm-based genre recognition.

Confusion Matrix (%)

	Rock	Hi-Hop
Rock	76	24
Hi-Hop	20	80

Fig. 7. 32-State Discrete Observation HMM Results

Confusion Matrix (%)

	Rock	Hi-Hop
Rock	74	26
Hi-Hop	22	78

Fig. 8. 16-State Discrete Observation HMM Results

Confusion Matrix (%)

	Rock	Hi-Hop
Rock	80	20
Hi-Hop	18	82

Fig. 9. 4-State Discrete Observation HMM Results

C. Synthesis

Recognizable synthesis was achieved and can be seen by comparing the original energy, time-domain, and spectrogram data to the optimal state-sequence synthesized data. The synthesis, however, proved quite difficult to control using the simplistic synthesis model shown in fig. 4. Additionally, the limited frequency range of the MFCC coefficients severely limited the quality of the synthesized sounds. Unfortunately, increasing the feature vectors limits the overall accuracy of HMM model due to the limited size of data. Moreover, the MFCC data does little to model time-domain information which is critical for modeling the transient behavior of drum signals. An attempt to integrate the delta and delta-delta MFCCs was made with little success due to the limited database size. Increasing the dataset size and adding significant features such as time-domain volume envelope information should significantly increase the behavior for synthesis. See figures 10, 11, 12 below for an example of a synthesized hi-hat drum sound (see http://ccrma.stanford.edu/∼njb/cs229 for sound examples). The state-sequence transitions can be visibly seen through the abrupt changes in each plot of data over time. When comparing sound quality between the isolated instrument synthesis sounds, the hi-hat and snare instruments proved the most effective because of the more noise modeled sound. The tom drum and bass drum proved to be more difficult and require a refined excitation signal.

VII. Conclusions

Overall, satisfactory results were achieved. Isolated recognition and rhythmic transcription or sequential isolated recognition for genre classification illustrated significant promise with little future refinement making search-by-rhythm a possibility for future revision. Unfortunately, the HMM synthesis proved to be quite difficult and exhibited a large need for improvement with respect to the signal processing synthesis model. Moreover, increasing the dataset size will be needed and should significantly improve the synthesis model as well as the isolated recognition.

Acknowledgment

Thank you to Professor Andrew Ng, the CS229 course TAs, and Professor Ge Wang of the Center for Computer Research in Music and Acoustics (CCRMA) for advice and guidance.

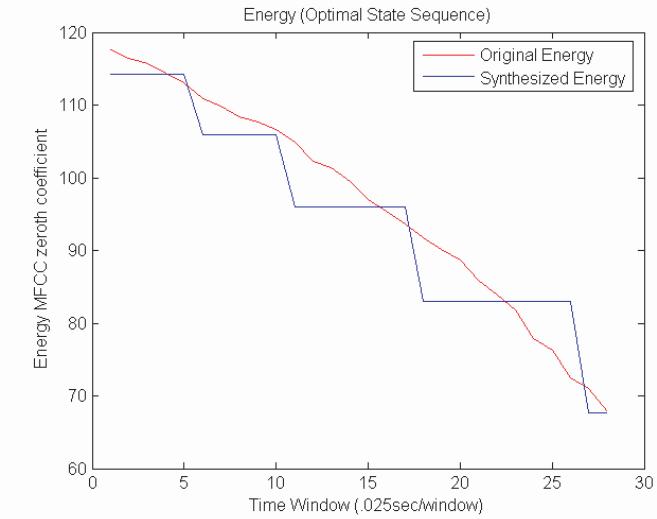

Fig. 10. Original vs. Synthesized Optimal State Sequence Energy

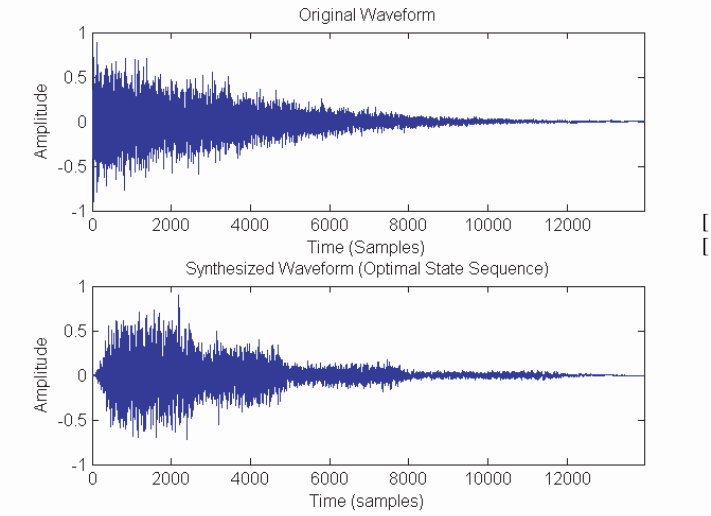

Fig. 11. Original vs. Synthesized Optimal State Sequence Time Domain Signal

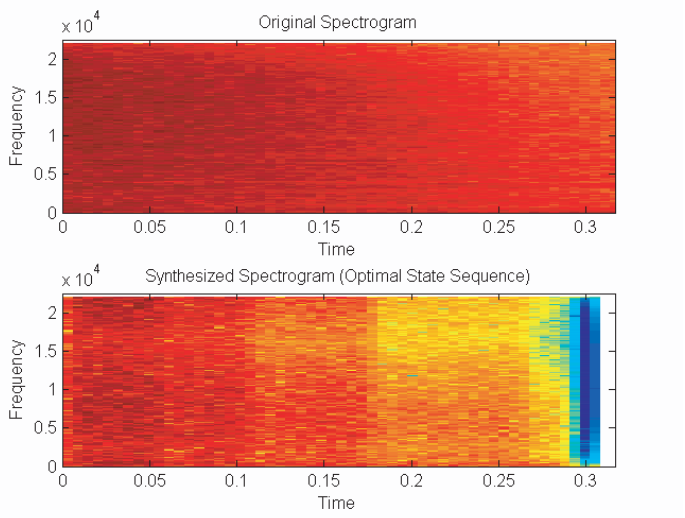

Fig. 12. Original vs. Synthesized Optimal State Sequence Spectrogram

REFERENCES

[1] D. FitzGerald and J. Paulus, "Unpitched percussion transcription," in *Signal Processing Methods for Music Transcription*, A. Klapuri and M. Davy, Eds. Springer-Verlag, 2006, pp. 131–162.

[2] D. Ellis, "Rasta/plp/mfcc feature calculation and inversion." [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/

[3] P. Davis, S.; Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, Aug 1980.

[4] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practices*. New Jersey: Prentice Hall, 2001.

[5] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proceedings of the First International Symposium on Music Information Retrieval (ISMIR)*, Plymouth, Massachusetts, oct 2000. [Online]. Available: citeseer.ist.psu.edu/logan00mel.html

[6] "Jason mcgerr sessions refill." [Online]. Available: http://www.propellerheads.se/

[7] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[8] K. Murphy, "Hidden markov model (hmm) toolbox for matlab," 1998. [Online]. Available: http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

[9] J. Paulus and A. Klapuri, "Conventional and periodic ngrams in the transcription of drum sequences," 2003. [Online]. Available: citeseer.ist.psu.edu/paulus03conventional.html

[10] t. . Yamagishi, J .

[11] A. Tokuda, K.; Heiga Zen; Black, "An hmm-based speech synthesis system applied to english," *Speech Synthesis, 2002. Proceedings of 2002 IEEE Workshop on*, pp. 227–230, 11-13 Sept. 2002.

Confusion Matrix (%)

	s	b	c	h	t	si	sb	sc	sh	st	bc	bh	bt	ch	ct	ht	sbc	sbh	sbt	bch	bct	cht	sch	sct	sht	bht	sbch	sbct
s	**86**	0	0	0	0	0	11.5	0.5	1	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0.5	0	0	0
b	0	**80**	0	0	2	0	11.5	0	0	0	0	0	5.5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
c	0	0	**76.5**	0	0	0	0	1.5	0	0	1.5	0	0	17	0	0	0.5	0	0	0	0	0	2.5	0.5	0	0	0	0
h	2	0	0	**80.5**	0	0	5	0	10	0	0	0.5	0	0.5	0	0	0	1.5	0	0	0	0	0	0	0	0	0	0
t	0	0	0	0	**76**	0	0	0	0	0.5	0	0	23	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0
si	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sb	0.5	19.5	0	0	0	0	**70**	0	0.5	0.5	0	2	0	0	0	0	0	2.5	3.5	0	0	0	0	0	1	0	0	0
sc	11.5	0	11.5	0	0	0	0.5	**48.5**	4.5	0	0.5	0	0	2.5	0	0	3	0.5	0	0	0	0.5	13.5	1.5	0	0	1.5	0
sh	12	0	0	7	0	0	2	3	**59.5**	0	0	2	0	1.5	0	0	0	6.5	0	0	0	0	1	0	2.5	0.5	2.5	0
st	0.5	0.5	0	0	4.5	0	6	0	1	**44**	0	0	5	0	0	0	0	1	34	0	0.5	0	0	0	2.5	0	0	0.5
bc	0	0	0	0	0	0	0	0	0	0	**49.5**	0	0	0	8.5	0	20.5	0	0	9.5	3.5	1	0.5	3	0	0	2.5	1.5
bh	0	1	0	0	0	0	7.5	0	0	0	0	**61.5**	0	0	0	2	0	12.5	0	4	0	0	0	0	1.5	9.5	0.5	0
bt	0	3	0	0	22.5	0	0	0	0	0.5	0	0	**72**	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
ch	0	0	18	4.5	0	0	0	2	3.5	0	1	0	0	**40**	0	0	2	0	0	4.5	0	0	20.5	0.5	0	0	3.5	0
ct	0	0	0	0	0	0	0	0	0	0	7	0	0	0	**23**	0	8.5	0	0	0	23.5	3	0	23.5	0	0	0.5	11
ht	0	0	0	0	1	0	1.5	0	0	3.5	0	3	1	0	0	**39.5**	0	2	1.5	0	0	1	0	0	11.5	33.5	1	0
sbc	0	0	0	0	0	0	0	2	0	0	25	0	0	0.5	2.5	0	**39.5**	0	0	3.5	0	1.5	3.5	9.5	1	0	8	3.5
sbh	1	0	0	0	0	0	14	0.5	3.5	0	0	11.5	0	0	0	1	0	**47**	0.5	1.5	0	0	0.5	0	8.5	5	5.5	0
sbt	0	0.5	0	0	1	0	7	0	0	23	0	0	10.5	0	0	0	0	3	**53.5**	0	0	0	0	0	1.5	0	0	0
bch	0	0	0	0	0	0	0	0	1.5	0	12	6	0	2	1	0	11.5	3	0	**38.5**	0.5	1	1	1	0	0.5	19	1.5
bct	0	0	0	0	0	0	0	0	0	0	7	0	0	0	23	0	8.5	0	0	0	**23.5**	3	0	23.5	0	0	0.5	11
cht	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	7	3.5	3.5	1.5	0	3	6.5	**30.5**	0.5	11	4.5	5.5	5	17.5
sch	0	0	6	2	0	0	0	9	7	0	0	0	0	21	0	0	4	0	0	1	0	0.5	**41.5**	0.5	0.5	0	7	0
sct	0	0	0	0	0	0	0	1	0	1	3.5	0	0	0	22	0	18.5	0	0	0	8.5	6	0.5	**24.5**	0.5	0	3	11
sht	0	0.5	0	0	0	0	4.5	0	1.5	8.5	0	3.5	0	0	0	15	0	7.5	4	0.5	0	2	0	0	**34.5**	18	0	0
bht	0	0.5	0	0	0	0	0	0	0.5	3	0	11	2	0	0	26	0	7.5	1.5	1	0	1	0	0	9.5	**36**	0.5	0
sbch	0.5	0	0	0	0	0	0	1	3	0	4.5	3	0	2.5	0	0	18	6	0	24	0	1.5	7.5	1.5	2	0	**25**	0
sbct	0	0	0	0	0	0	0.5	0	0	0	1.5	0	0	0	12.5	1	3.5	0	1	0.5	34	8	0	16	0	0.5	1	**20**

Fig. 13. All Class Confusion Matrix (%)