POLYPHONIC MUSIC NOTE ONSET DETECTION USING SEMI-SUPERVISED LEARNING

Wei You and Roger B. Dannenberg
Carnegie Mellon University
Schools of Computer Science and Music

ABSTRACT

Automatic note onset detection is particularly difficult in orchestral music (and polyphonic music in general). Machine learning offers one promising approach, but it is limited by the availability of labeled training data. Score-to-audio alignment, however, offers an economical way to locate onsets in recorded audio, and score data is freely available for many orchestral works in the form of standard MIDI files. Thus, large amounts of training data can be generated quickly, but it is limited by the accuracy of the alignment, which in turn is ultimately related to the problem of onset detection. Semi-supervised or bootstrapping techniques can be used to iteratively refine both onset detection functions and the data used to train the functions. We show that this approach can be used to improve and adapt a general purpose onset detection algorithm for use with orchestral music.

1 INTRODUCTION

Finding the beginning of notes, or *note onsets*, in music audio is a problem that is widely studied. By finding note onsets, we can segment continuous music into discrete note events, benefiting tempo estimation, beat finding, automatic music transcription, and other analysis tasks. These in turn are often used as components in systems for music indexing and retrieval, music fingerprinting, and music similarity. Thus onset detection is a fundamental task for music information retrieval. However, because of the variability within and between musical instruments, finding note onsets is not trivial. In polyphonic pieces, note onsets may be difficult to separate from other notes, and in large ensembles such as an orchestra, masses of note onsets can be difficult to handle.

Our focus is on massively polyphonic music, *e.g.* orchestra music. One reason previous work has focused on monophonic and polyphonic piano music is the availability of test data. Hand labeling music onsets is tedious work, and it would be very expensive to label all note onsets in large polyphonic works. For example, Beethoven's Symphony no. 5 in C minor, first movement, has more than 10,000 notes and over 2,000 separate onset times over a duration of about 435s. If we assume labeling one note takes 1 minute (quite optimistic in our experience), then more than 30 hours will be needed to label one movement.

To solve this dilemma, *audio-to-score alignment* is used to estimate note onsets automatically. Typically, score alignment is performed with chroma features, which summarize 50 to 250ms windows of audio, but this does not provide the high time resolution one would like for onset labeling. For example, it is not unusual to see an *average* inter-onset time of 200ms in a polyphonic work. Therefore, we use a *semi-supervised learning method* to enhance the performance of the audio-to-score alignment, leading to improved onset detector training [6]. Thus, the task of acquiring training data is simply to locate corresponding audio and MIDI files and run a relatively fast alignment algorithm. Vastly increased numbers of training examples improve functions for onset detection.

We compare the results of our trained onset detector to a high-quality, open-source onset detector, Aubio (http://aubio.piem.org). To measure only improvements due to semi-supervised training, we used exactly the same features as Aubio, and we also used the Aubio peak-picking algorithm for our system. We also tried adding new features, hoping that machine learning would be able to take advantage of the additional information.

After the related work section that follows, we explain the techniques used by our onset detection system. Section 4 presents an evaluation. Our findings are discussed in Section 5, which is followed by a concluding section.

2 RELATED WORK

Current practice in note onset detection can be separated into two general approaches: 1. Apply a detection function, often based on change in spectrum and overall power; then use a temporal peak-picking algorithm to find local maxima in output from the detection function [2, 4, 1, 3]. 2. Using a variety of features, apply machine-learning techniques to build a note onset classifier.

Kapanci and Pfeffer [8] use a Hierarchical Model and support vector machine to estimate whether an onset is within a span of time. Marolt, et al., and Lacoste and Eck [10, 9] use Neural Network approaches for onset detection. Dannenberg and Hu [6] used semi-supervised (or bootstrap) learning with Neural Networks for monophonic and piano music onset detection.

Many of the research systems using machine-learning

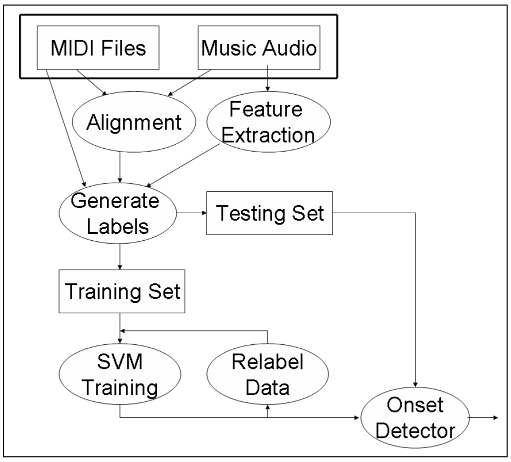

Figure 1. Semi-supervised learning of an onset detector.

techniques are trained on the data of the Music Information Retrieval Evaluation eXchange (MIREX), which consists of 30 solo drum, 30 solo monophonic pitched instrument, 10 solo polyphonic pitched instrument, and 15 complex (much less complex than orchestra works) pieces. The total length of all sets is 14 minutes. This dataset is small for training a polyphonic music onset detector.

3 TECHNIQUES

Our approach acquires training data using score alignment to match symbolic (MIDI) data to recordings of acoustic music performances [11]. (See Figure 1.) The MIDI data contains the onsets while the alignment tells us where to find these onsets in the audio recordings. Onset labels are used as training data for a Support Vector Machine, and the output is used to further refine the alignment data. This bootstrapping process is iterated until it converges. The final onset detector is treated with adaptive thresholding and peak picking to estimate note onset locations. These steps are covered in more detail below.

3.1 Audio-to-Score Alignment

Audio-to-score alignment uses chroma vector features [12] and dynamic programming to align audio to note data from a standard MIDI file [5]. The chroma vector captures information about harmony and melody during a short time interval, typically 50 to 250ms.

Alignment is performed by constructing a distance matrix S where $S_{i,j}$ is the Euclidean distance between audio chroma vector i and MIDI chroma vector j. Then, dynamic programming is used to find a path from $S_{0,0}$ to $S_{N,M}$ that minimizes the sum of distances traversed by the path. This path is then smoothed to form a continuous mapping between audio and MIDI. Each note onset time in the MIDI data can now be mapped to a corresponding time in the audio.

3.2 Acoustic Features

Feature selection is important for note onset detection. The main features used in our models are those of Aubio. These are *Energy of the Frame*, *High Frequency Content*, *Spectral Flux*, *Phase Deviation*, *Kullback Leibler Divergence*, *Modified KL*, and *Complex Domain* (see [2, 3] and http://aubio.piem.org for details.)

For some tests, we extended or modified the Aubio feature set with:

- *Higher order differences*: many features already include some measure of change, such as spectral flux. We added first- through fourth-order differences between features, expanding 7 features to 35 features per frame.

- *Larger frame*: Aubio uses a default window size of 1024 and a hop size of 512. We trained another onset detector by adjusting the window size to 16384 with a hop size of 2048 in order to capture change over longer time scales.

- *Chroma Flux*: We added a measure of change in the chroma vector, hoping to capture changes in melody and harmony, especially in string ensembles where slow onsets might not exhibit typical onset features.

3.3 Semi-supervised learning

From score alignment, we obtain a large set of labeled training data, but the labels are based on rather large windows, and the chroma vector features are chosen more for gross alignment than for precise onset detection. A bootstrapping technique improves the labels while simultaneously learning a good onset detector [6].

We use a Support Vector Machine (SVM) classifier with Radial Basis Function (RBF) kernels. Two parameters need to be determined before using RBF kernels: C and γ. It is not known beforehand which C and γ are best for the classification problem, but the difference in classification accuracy between a good pair of (C, γ) and a bad one can be huge. Therefore, parameter searching should be done before training the whole model.

We used the LIBSVM library (http://www.csie.ntu.edu .tw/ cjlin/libsvm/) for the implementation of SVMs in our learning. Before training on the whole dataset, we randomly choose several independent subsets from the whole dataset, and then apply grid-search on C and γ using cross-validation. Then, we train classifiers on the whole dataset using the best parameter pairs and choose the one with the best performance.

The training set is initialized using features from music audio. The onset time of the k^{th} onset of the aligned score is denoted by T_k. A frame is labeled 1 (onset) if its time matches some onset time T_k, and 0 (no onset) otherwise. The SVM is trained on this data, producing an onset detector whose per-frame output is interpreted as a probability (from 0 to 1) of an onset in the frame.

Music Title	Onset frames
Beethoven, Sym. #5 Op. 67, 1st mvt.	2377
Bach, Brandenburg #5, BWV 1050, 1st mvt.	4511
Chopin, Pn. Concerto #1, Op. 11, 2nd mvt.	3146
Haydn, Sym. #94, 1st mvt.	4465
Mozart, Vn. Concerto #5, K. 219, 1st mvt.	3504
Mozart, Pn. Sonata K. 331, 1st mvt.	2349
Mozart, Sym. #40, K. 550, 1st mvt.	2076
Mozart, Cl. Quintet, K. 581, 1st mvt.	2579
Bach, Passacaglia & Fugue, BWV 582	3375
Tchaikovsky, Sym. #6, Op. 67, 1st mvt.	1638

Table 1. Training data.

The training data is then relabeled as follows: First, a per-frame probability density $P(i)$ is estimated by initializing each $P(i)$ to a small constant. Then, for each onset time T_k, as predicted by the score alignment, add a Gaussian window with mean T_k and standard deviation of about 100ms to P. Compute $f(i) = P(i) \times O(i)$ where $O(i)$ is the output of the trained onset detector (so far), and $f(i)$ represents onset probability after considering the prior knowledge P and the probability based on features $O(i)$. Finally, for each onset time T_k, find the largest frame value $f(i)$ within a window W_1 and (re)label it as 1; all other frames are labeled 0. The W_1 window size is defined as follows:

$$\begin{aligned} W_1(i) = [&max((T_k + T_{k-1})/2, T_k - W), \\ &min((T_k + T_{k+1})/2, T_k + W)] \end{aligned} \quad (1)$$

where W is 250ms. Retrain the SVM classifier with the new labels until the recall on a set of hand-labeled onsets stops increasing. The whole learning process usually takes 8 to 10 iterations.

The resulting onset detector is good, but returns many false positives. The detector can be further improved by applying a peak-picking algorithm to its output. We use the same peak-picking algorithm as in Aubio to simplify comparisons. It works by comparing the onset detection function output to an adaptive threshold, then searching for local maxima when the threshold is crossed.

4 EVALUATION AND RESULTS

We chose ten polyphonic music pieces as the training data, listed in Table 1. All pieces come from the RWC Music Database [7]. The whole length of the dataset is more than 90 minutes. There are 76,028 notes, with 30,020 distinct onset times separated by at least one frame period. In the training set, there are 30,020 instances labeled as positive; all the rest are negative.

The testing set consists of 18,521 notes, with 3,225 distinct onset times, taken from Johann II Strauss's Blue Danube. As a reference, we compared our note-onset detector performance to that of Aubio. Figure 2 plots Precision-Recall curves for various configurations, which differ only in the features used. The best F-measures ($F = 2PR/(P+R)$, where P is precision and R is recall) are *Aubio*: 0.38, *SVM*: 0.47, *Long Frame*: 0.32, *+Chroma*: 0.44, *+Higher Order*: 0.44, *MIDI*: 0.40, *NoBootstrap*: 0.03 (these labels are defined and also used in Figure 2). Except for the *Long Frame* features, the onset detectors trained with semi-supervised learning out-performed the Aubio onset detector. Interestingly, the original Aubio feature set worked better than any of our alternatives.

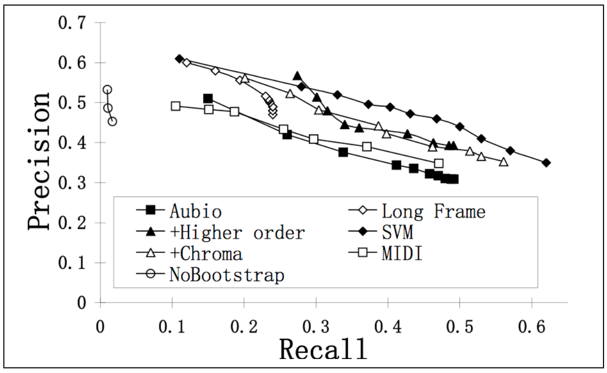

Figure 2. Performance comparison using Precision-Recall curves. *Aubio*: hand-tuned onset detector, *SVM*: trained onset detection function using Aubio features, *Long Frame*: same as *SVM* except larger window size, *+Chroma*: same as *SVM*, but chroma flux feature is added, *+Higher Order*: same as *SVM*, but higher-order difference features are added, *MIDI*: trained on audio synthesized from MIDI files, *NoBootstrap*: trained on 100 hand-labeled onsets, no semi-supervised learning.

To give some idea of computation time, a typical run labels 564s of audio containing 3225 onsets. The total computation time is 433s, of which 30s is spent calculating Aubio features, 385s for SVM classification, and 18s for peak picking. It takes about 70 hours to train the SVM classifier on a 2.4GHz Intel P4 system, including the 8 to 10 bootstrapping iterations.

5 DISCUSSION

In this study, machine learning outperformed a hand-tuned detection system on our test data, indicating that our semi-supervised learning approach is successful. At first, we thought that Aubio would not perform particularly well on orchestral music and that we would obtain significant improvements by introducing new features. We thought of machine learning as an efficient way to explore various new features. Our data suggests that finding new features or tuning them to orchestral music may not be so simple. None of the features we added offered significant improvement to the original feature set we took from Aubio. The improvement must be attributed to the improved classifier function.

Whether improvements come from new features or refining the onset prediction function, the bottom line is that our improvements are a direct result of machine learning, and any machine learning approach depends upon large amounts of training data. Our semi-supervised learning approach offers a working solution to this problem. We did not perform an extensive comparison of our onset detector to Aubio across a wide selection of music. Our only claim is that semi-supervised learning can automatically adapt an onset detector to a class of music (in this case massively polyphonic orchestral music) with good results.

To further study the contributions of semi-supervised learning, we trained using SVM with a small set of 100 hand-labeled onsets and with a large set of onsets from synthesized MIDI files (see Figure 2). We also tried training on the score-alignment data without bootstrapping. The results show it is important to have both a large data set *and* actual acoustic data.

Our test data contains thousands of points that seem typical of orchestral music onset detection, but all of these come from one performance that was held out from training. Further testing has begun to confirm these initial results, and a full cross-validation study is in progress.

Analysis of our audio-to-score alignment using a random sample of 100 hand-labeled onsets reveals that there are significant alignment problems in some cases. Further work is needed to identify the sources of these problems, which could include errors in MIDI files and limitations of our alignment algorithm. When alignment errors are more than about 100ms, the onset labels might as well be random because the average interval between onsets in our entire dataset is about 200ms. That we are able to show good performance in spite of the fact that many of our training data points are effectively random is actually a strong endorsement of our approach. It seems likely that the bootstrapping process is "searching" for real onsets in the neighborhoods of erroneously labeled frames, minimizing the damage of the bad data. Our test data is based solely upon score alignment without further refinement using bootstrapping techniques. We are working to characterize and improve the quality of the test data.

6 SUMMARY AND CONCLUSIONS

Onset detection in polyphonic music and particularly orchestral music is difficult. We explored the use of machine learning to improve onset detection functions. To solve the problem of training data, we use a semi-supervised learning technique combined with score alignment. The result of alignment is an estimate of the onset time of every note in the MIDI file, and these estimates are improved by iteratively applying our onset detector and then retraining on the new data.

Our resulting onset detection function shows a significant improvement over a hand-tuned onset detector using the same features. Our onset detector is trained on polyphonic music that is mostly from orchestra performances. While this is an interesting problem in itself, future work might explore whether it is better to specialize onset detectors for different types of music or to pool all available training data and create one general-purpose onset detector. Either way, semi-supervised learning is a promising approach to gathering large amounts of training data, leading to significant improvements in onset detection.

7 REFERENCES

[1] Alonso, M., Richard, G., and David, B. "Extracting note onsets from musical recordings," *IEEE International Conference on Multimedia & Expo*, Amsterdam, 2005.

[2] Bello, J., Duxbury, C., Davies, M., and Sandler, M. "On the use of phase and energy for musical onset detection in the complex domain," *IEEE Signal Processing Letters*, vol. 11, no. 6 June 2004.

[3] Brossier, P., Bello, J., and Plumbley, M. "Real-time temporal segmentation of note objects in music signals," in *Proceedings of the ICMC*, Miami, Florida, 2004, pp. 458-461.

[4] Collins, N. "A Change Discrimination Onset Detector with Peak Scoring Peak Picker and Time Domain Correction," *Music Information Retrieval Exchange*, MIREX 2005.

[5] Dannenberg, R., and Hu, N. "Polyphonic Audio Matching for Score Following and Intelligent Audio Editors," in *Proceedings of the 2003 Inter. Computer Music Conference*, Singapore, 2003, pp. 27-34.

[6] Dannenberg, R., and Hu, N. "Bootstrap learning for accurate onset detection." *Machine Learning* 65 (2-3), pp. 457-471.

[7] Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R. "RWC Music Database: Popular, Classical, and Jazz Music Databases," in *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, (October) 2002, pp. 287-288.

[8] Kapanci, E., and Pfeffer, A. "A hierarchical approach to onset detection," in *Proceedings of the ICMC*, Miami, Florida, 2004, pp. 438-441.

[9] Lacoste, A., and Eck, D. "Onset Detection with Artificial Neural Networks for MIREX 2005," *Music Information Retrieval Exchange*, MIREX 2005.

[10] Marolt, M., Kavcic, A., and Privosnik, M. "Neural networks for note onset detection in piano music," in *Proceeding of the 2002 ICMC*, Miami, Florida, 2002.

[11] Turetsky, R., and Ellis, D. "Ground-Truth Transcriptions of Real Music from Force-Aligned MIDI Syntheses," *4th Inter. Symposium on Music Information Retrieval ISMIR-03* Baltimore, 2003, pp. 135-141.

[12] Wakefield, G. "Mathematical representation of joint time-chroma distributions," in *International Symposium on Optical Science, Engineering, and Instrumentation, SPIE'99*, Denver, 1999.