

Automatic Extraction of Tempo and Beat from Expressive Performances (Dixon)

Naru Sundar, ISE 575

This paper involved an algorithm for detecting tempo and beats from audio input consisting of either raw audio data (PCM) or symbolic data (MIDI). The paper discussed some history of the various techniques involved, and then continued to describe the algorithm in detail, concluding with results from applying the algorithm on various types of inputs. The algorithm consisted of an initial phase wherein the input data was converted to note onset data and salience information. Onset intervals were clustered to create tempo hypotheses. Each hypothesis was then tested via an agent that kept track of how well the hypothesis predicted the input events. The highest scoring agent returned the tempo.

The initial phase of the algorithm required the conversion of input data to note onset information. If the input data was symbolic in nature this onset information was available immediately. For pure audio inputs further processing was necessary. After filtering and smoothing the waveform, a peak-picking algorithm was used to identify peaks as onset events. I found this to be the weakest part of the algorithm, since there is a lot of room for onset information to be lost. The authors themselves admitted this part of the algorithm was not general enough.

The next phase consisted of considering all possible pairs of note intervals. The pairs were clustered together such that note intervals close to the average interval of events in a cluster were added to that cluster. Clusters whose averages drifted together were combined. The clusters were ranked by a score that took into account both the number of events in the cluster as well as the size of clusters whose note intervals were (within tolerance) an integer multiple of that cluster's interval.

The last phase consisted of constructing agents who used each cluster's interval as a tempo predictor. The agent processed events one at a time for each cluster, accumulating the event into the cluster's event history if the cluster accurately predicted that event as being part of the "beat." If an agent completely failed to predict an event it was removed from the list of agents. If an agent predicted an event within a tolerance that was close enough not to discard, but not close enough to count as a prediction, then the agent was cloned, with the old agent containing the event in its history while the new agent did not.

Each agent accumulated a score based on the error margin of the current event versus the prediction. An optional part of the score computation involved a notion of note salience. The idea was that notes whose pitch, duration and amplitude indicated a particular importance in the metric, should weight the agent's score higher if predicted. The salience function was computed as a linear function of these three parameters, with constants determined empirically. A multiplicative function was proposed but not used in the experiments.

The results indicated overall success. Various forms of input from simple pop songs to complicated syncopated jazz songs were used. Complex syncopation were more difficult, but not out of the range of ability of the algorithm. Note salience proved to have a positive effect, as the experiments which tested the results with and without note salience showed. In the main I found the algorithm an average one, but not ideal. The algorithm was clearly not suited for real-time purposes, and I would have proposed a much larger scheme of tests involving a wider variety of inputs. Classical music of various types exist in easily obtainable midi form, as is true for many pop songs. This large gamut of inputs would help show the weaknesses and strengths of the algorithm. Pushing the algorithm towards a more real-time form would also be desirable for any practical applications.