Extensions on Rhythmic Quantization from Inter-Onset Intervals

Sarah Shader with advisor Eran Egozy

October 4, 2017

1 Introduction

Music transcription has been an area of interest for humans for quite a while. By hearing a piece of music performed, one wants to write (transcribe) this in a format that can be used to later reproduce the same piece of music, typically in the form of sheet music. As more people apply computer science to music, there is a greater demand for musical transcription in a format that makes sense for a computer. Since music transcription is hard and time consuming for humans, automatic music transcription has become an active area of research.

Most music can be broken down into pitch and rhythm. While both of these are interrelated, and there has been research that attempts to transcribe both in parallel, in my proposed project, I plan to focus on rhythm transcription. Rhythm transcription, much like general music transcription, can be a hard problem even for humans due to fluctuating tempos, artistic interpretation of the written rhythm. Essentially, almost no musical performance will be played with perfect robotic rhythmic precision. This makes the problem of rhythmic transcription even more challenging for computers, since this requires extracting the underlying structure behind inherently imperfect data.

Rhythmic transcription can be broken down further into four parts: onsets, tactus, tempo, and quantization [1]. Detecting onsets consists of converting an audio recording to a set of MIDI onset times that correspond to the start time of each note played. Since this itself is an area of active research, in this project, I will assume that this has already been done, and we have as input a set of inter-onset intervals (IOIs). Tactus refers to the beat that one would tap his or her foot to, and relatedly tempo refers to the rate of these beats. Finally, quantization, which is the main focus of this project, refers to mapping rhythmically imperfect IOIs to musical quantities such as half notes, quarter notes, triplets, etc.

Tempo and quantization are closely linked, since tempo clearly affects the length of IOIs corresponding to a particular note, but also knowing the correct quantizations indicates what tempo the notes were played at. For this reason, several papers have created systems that transcribe tempo and quantization simultaneously [2], [4], [6]. Most of these papers involve creating a Bayesian network to model the dependencies between tempo and quantized note length, and then using some sort of inference algorithm to make computing the most probable result according to the model feasible. I propose to first implement one of these rhythm quantization models, and then as an extension, do one of three things, each of which I will discuss in the following sections.

2 Finding a Transition Matrix for Rhythm Quantization

All of these systems rely on having an accurate transition matrix that indicates the probability of a certain note length given the previous (e.g. it is more likely that a single triplet is followed by another triplet than a quarter note). In [2] in fact shows how important a good transition matrix is for accurate rhythm transcription. The most common approach to creating a transition matrix is to take the probabilities found in already transcribed music similar to the piece that is being transcribed. However, this creates a bit of a circular problem, where you can only really get an accurate transcription if you already have several accurate transcriptions for similar pieces. Another paper [3] use a heuristic relating to bit complexity to populate this transition matrix. This leverages the intuition that it is much more likely for a note to end half way through a measure than it is to end $\frac{15}{32}$ of the way through a measure.

This second approach offers much more flexibility than the first approach, since this does not depend on already having accurate transcriptions of similar pieces. If the second approach can transcribe rhythm as accurately as the first, this would provide a more general algorithm for rhythm detection. In general, it would be interesting to investigate how to best find a transition matrix that performs well on a large scope of pieces. Aside from comparing various heuristics for this transition matrix, it would be interesting to see if an algorithm could learn a good transition matrix by way of machine learning. While this approach does still rely on having other similar pieces to create a transition matrix, it would be automating this process, and could be less dependent on having a completely accurate transcription of the other pieces. A machine learning approach to creating the transition matrix could also eliminate any overfitting and produce a more general transition matrix that could apply to a broader spectrum of pieces.

3 Rhythm Quantization from Multiple Samples

The most common way to evaluate the performance of a rhythm quantization algorithm is to give as input a specific song, and see how many mistakes the algorithm makes. Some papers do this with several different versions of the same song to see if the algorithm can accurately transcribe regardless of the artistic interpretation of the performer. However, there has been little research on algorithms that take in multiple performances of the same piece and attempt to use all of them to come up with a more accurate rhythm quantization. Even among one performer, playing the same piece repeatedly, there are small variations in tempos and note lengths from performance to performance. It seems likely that by having multiple, slightly varied, performances of the same piece, an algorithm could be adapted to detect the same underlying structure of all the performances.

Clear approaches would be to average the IOIs, and hope that a performer plays close to the true rhythm on average. However, this would not work well in the presence of lots of tempo variation. Alternatively, the same rhythm quantization algorithm can be run separately on all the performances, and any note quantizations that are common among all the performances are almost surely correct. This approach is less straightforward than simply averaging the IOIs, since it is unclear how to deal with cases where different intervals are classified as different note types. It is also possible to develop more sophisticated approaches. Perhaps to fix the issues with tempo variation for averaging IOIs, one could detect the tempos of all the performances separately, and then scale all the performances by tempo before averaging. If one could achieve much more accuracy by simply collecting a few more performances of the desired piece, this would make it much easier to quickly get accurate rhythmic transcriptions.

4 Supervised Rhythm Quantization

Inevitably, these models for rhythm quantization will make mistakes and mis-quantize a few notes. Due to the interdependent nature of these models, oftentimes correcting just one error will cause other errors to be corrected when the algorithm is run again. In [2], ~~they~~ plot the number of errors with no error correction, the number of errors after correcting one error, the number of errors after correcting two errors, etc. and demonstrate that the rhythm quantization gets significantly better by simply correcting a couple errors. This paper performs the error correction by simply correcting the first error. However, it might make sense to have a more sophisticated strategy for error correction to allow a user to answer a few questions to supervise the rhythm quantization process.

All of these models have some probability associated with each note quantization, and the best note quantization is the maximum a posteriori (MAP) estimate, which is the most likely configuration of note quantizations and tempos given the observed set of IOIs. Within the MAP estimate, there are note quantizations that have higher probabilities associated with them compared to others. These probabilities can be viewed as how confident the algorithm is in its classification, so correcting these errors, as opposed to just the first error to occur, could potentially produce a better overall rhythm quantization. It also stands to reason that there may be two estimates that have similar probabilities, so by simply returning both to the user, and allowing the user to pick the one that sounds “more correct” could be an improvement on returning the single MAP estimate, and making the user manually correct errors.

References

[1] Klapuri, A. P. (2004). Automatic Music Transcription as We Know it Today. *Journal of New Music Research*, 33:3, 269-282.

[2] Raphael, C. (2001). Automated Rhythm Transcription. *ISMIR 2006 - 7th International Conference on Music Information Retrieval.*

[3] Cemgil, A. T., Kappen, B. (2003). Monte Carlo Methods for Tempo Tracking and Rhythm Quantization. *Journal of Artificial Intelligence Research* 18, 45-81.

[4] Whiteley, N., Cemgil, A., Godsill, S. (2006). Bayesian Modelling of Temporal Structure in Musical Audio. *ISMIR 2006 - 7th International Conference on Music Information Retrieval.* 29-34.

[5] Nakamura, E., Yoshii, K., Sagayama, S. (2017). Rhythm Transcription of Polyphonic Piano Music Based on Merged-Output HMM for Multiple Voices. *IEEE/ACM TASLP*.

[6] Nakamura, E., Itoyama, K., Yoshii, K. (2016). Rhythm Transcription of MIDI Performances Based on Hierarchical Bayesian Modelling of Repetition and Modification of Musical Note Patterns. *Proc. EUSIPCO* 1946? 1950.