LARGE-SCALE CONTENT-BASED MATCHING OF MIDI AND AUDIO FILES

First author
Affiliation1
author1@ismir.edu

Second author
Retain these fake authors in submission to preserve the formatting

Third author
Affiliation3
author3@ismir.edu

ABSTRACT

MIDI files, when paired with corresponding audio recordings, can be used as ground truth for many music information retrieval tasks. We present a system which can efficiently match and align MIDI files to entries in a large corpus of audio content without using any metadata. The core of our approach is a neural network-based cross-modality hashing scheme which transforms acoustic feature matrices and MIDI piano rolls into sequences of vectors in a common Hamming space. Once represented in this way, we can efficiently perform large-scale dynamic time warping searches to match MIDI data to audio recordings. We evaluate our approach on the task of matching a huge corpus of MIDI files to the Million Song Dataset.

1. TRAINING DATA FOR MIR

Central to the task of content-based Music Information Retrieval is the curation of ground-truth data for tasks of interest (e.g. timestamped chord labels for automatic chord estimation, beat positions for beat tracking, prominent melody time series for melody extraction, etc.). The quantity and quality of this ground-truth is often instrumental in the success of MIR systems which utilize it as training data. Unfortunately, creating appropriate labels for a recording of a given song by hand often requires person-hours on the order of the length of the song. This often arguably makes the available training data a bottleneck to success for a given content-based MIR task.

It has previously been observed that MIDI files, when time-aligned to corresponding audio recordings, can be used to infer ground-truth information about a given song [6, 13]. This is due to the fact that a MIDI files can be viewed simplistically as a timed sequence of note annotations or a piano roll. It is much more straightforward to estimate, e.g., beat locations, chord labels, and the predominant melody from these representations than one which was derived from an audio signal. Unsurprisingly, a handful of tools have been developed for inferring this information from MIDI files [4, 5, 9, 10].

In [7], it is argued that some of the biggest successes in machine learning are thanks to the fact that "...a large training set of the input-output behavior that we seek to automate is available to us in the wild." The main motivation behind this project is that this crucial availability of data holds true for MIDI files - through a large-scale web scrape, we obtained 250,000 unique MIDI files, which is orders of magnitude larger than the datasets typically used for MIR research. We believe this proliferation of data is largely caused to two factors: First, that karaoke files are typically distributed as MIDI data and that karaoke is wildly popular, and second, that transcribing popular music as MIDI files was a common pastime for hobbyist musicians in the nineties.

1.1 Wrangling MIDI Files

The mere existence of a large collection of MIDI data is not enough, however. As noted above, in order to use MIDI files as ground truth, they need to be both matched (paired with a corresponding audio recording of the same song) and aligned (adjusted so that the timing of the events transcribed in the file match the audio recording). The latter problem has seen a great deal of research effort [6, 13], and will not be a main focus of this work.

Given large corpora of audio and MIDI files, the task of matching entries from each may seem to be a trivial problem involving fuzzy text matching of the files' metadata. However, MIDI files have no formal mechanism for storing metadata (apart from text meta events, which are rarely used), and as a result the best-case scenario is that the artist and song title are included in the filename or subdirectory. While we found some examples of this in our collection of scraped MIDI files, the vast majority of the files had effectively no metadata information. Figure 1 shows a random sampling of subdirectory and filenames from our collection.

Fortunately, the goal of matching MIDI and audio files is to find pairs which have *content* in common (i.e., the MIDI file is a transcription of the audio file), an information source which is available regardless of metadata quality. However, comparing content has the potential to require much more computation than a fuzzy text comparison, and NM comparisons must be made to match a MIDI dataset of size N to an audio file dataset of size M. Motivated by this, we propose a system which can *efficiently* match MIDI files to audio based solely on their content. Our system learns to hash MIDI and audio content

```
J/Jerseygi.mid
V/VARIA18O.MID
Carpenters/WeveOnly.mid
2009 MIDI/handy_man1-D105.mid
G/Garotos Modernos - Bailanta De Fronteira.mid
Various Artists/REWINDNAS.MID
GoldenEarring/Twilight_Zone.mid
Sure.Polyphone.Midi/Poly 2268.mid
d/danza3.mid
100%sure.polyphone.midi/Fresh.mid
rogers_kenny/medley.mid
2009 MIDI/looking_out_my_backdoor3-Bb192.mid
```

Figure 1. Random sampling of 12 MIDI filenames and their parent directories from our corpus of 250,000 MIDI files scraped from the Internet.

to a common Hamming space where sequences of vectors can be compared efficiently using dynamic time warping (DTW).

The idea of using DTW distance to match MIDI files to audio recordings is not new. For example, in [8], MIDI-audio matching is done by finding the minimal DTW distance between all pairs of chromagrams of (synthesized) MIDI and audio files. Our approach differs in a few key ways: First, instead of using chromagrams (a hand-designed representation), we optimize a common representation for MIDI and audio data. This makes our system flexible with respect to the underlying feature representation used for each feature modality. Second, our datasets are many orders of magnitude larger (hundreds of thousands vs. hundreds of files), which necessitates a much more efficient approach. Specifically, by mapping to a Hamming space we greatly speed up distance matrix calculation and our proposed pruning techniques avoid a full DTW distance computation for the majority of file pairs.

In the following section, we detail the dataset of MIDI files we scraped from the Internet and describe how we prepared a subset for training our hasher. Our cross-modality hashing model is then described in Section 3. In Section 4, we cover our fast hash sequence DTW method and its accompanying pruning techniques. Finally, we evaluate our system's performance on the task of matching files from our MIDI dataset to entries in the Million Song Dataset [2].

2. THE MIDI DATASET

Our project began with a large-scale scrape of MIDI files from the Internet. We obtained 500,000 files, of which 250,000 were found to have unique MD5 checksums. Most of these files were transcriptions of pieces of music of varying duration and quality. As mentioned previously, the majority of these files had very little useful metadata information. However, we identified a subset of files where the subdirectory of the file indicated the artist and the filename indicated the song title. In order to normalize this metadata, we applied some manual text processing and resolved the artists and song titles against the Freebase [3] and Echo Nest [1] databases. This resulted in 17,000 MIDI files for about 9,000 unique songs. We will refer to this collection of files as the "clean MIDI subset".

The model we will use to hash MIDI and audio features to a common Hamming space (described in Section 3) requires a training set of feature vectors in each modality which should be mapped to similar hashes. In our application, we will be matching sequences of hashes, so we need to obtain pairs of sequences of feature vectors where the nth vector in one sequence should be mapped to the nth vector in the other. This necessitates a collection of MIDI files and audio recordings which we are confident are well-aligned in time. From this collection, we can extract features for each modality and be sure that there is a direct correspondence between the resulting hash sequences and the original feature vector sequences.

To obtain this training data, we combined three benchmark audio collections: CAL500 [14], CAL10k [11], and uspop2002 [1]. Each consists of a collection of music recordings, their metadata, and precomputed Echo Nest feature vectors (as used in the Million Song Dataset). To match these datasets to the clean MIDI subset, we performed a fuzzy matching of their metadata. We used Whoosh, a Python search engine package, to perform the fuzzy string search.

In order to obtain feature vectors from each modality which should have the same hash representation, we first need to align matched MIDI and audio files. We used the technique described in [12], which performs MIDI-audio alignment using DTW and reports a confidence score based on the normalized DTW distance. By listening to a random subset of 100 MIDI-audio alignments, we determined a confidence score threshold below which we could be reasonably certain that an alignment was successful. This resulted in about 5,000 aligned MIDI/audio pairs, of which 2,000 corresponded to unique songs.

3. CROSS-MODALITY HASHING OF MIDI AND AUDIO DATA

4. FAST DTW MATCHING OF HASH SEQUENCES

5. MATCHING MIDI FILES TO THE MSD

6. 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, 2004.

[2] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *ISMIR 2011: Proceedings of the 12th International Society for Music Information Retrieval Conference, October 24-28, 2011, Miami, Florida*, pages 591–596. University of Miami, 2011.

[3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD in-*

[1] http://developer.echonest.com/docs/v4

ternational conference on Management of data, pages 1247–1250. ACM, 2008.

[4] Michael Scott Cuthbert and Christopher Ariza. `music21`: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the 11th International Conference on Music Information Retrieval*, pages 637–642, 2010.

[5] Tuomas Eerola and Petri Toiviainen. MIR in Matlab: The MIDI toolbox. In *Proceedings of the 5th International Conference on Music Information Retrieval*, pages 22–27, 2004.

[6] Sebastian Ewert, Meinard Müller, Verena Konz, Daniel Müllensiefen, and Geraint A. Wiggins. Towards cross-version harmonic analysis of music. *IEEE Transactions on Multimedia*, 14(3):770–782, 2012.

[7] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

[8] Ning Hu, Roger B. Dannenberg, and George Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 185–188. IEEE, 2003.

[9] Cory McKay and Ichiro Fujinaga. jSymbolic: A feature extractor for MIDI files. In *Proceedings of the International Computer Music Conference*, pages 302–305, 2006.

[10] Colin Raffel and Daniel P. W. Ellis. Intuitive analysis, creation and manipulation of MIDI data with `pretty_midi`. In *Proceedings of the 15th International Conference on Music Information Retrieval Late Breaking and Demo Papers*, 2014.

[11] Derek Tingle, Youngmoo E. Kim, and Douglas Turnbull. Exploring automatic music annotation with "acoustically-objective" tags. In *Proceedings of the international conference on Multimedia information retrieval*, pages 55–62. ACM, 2010.

[12] Omitted to maintain anonymity.

[13] Robert J. Turetsky and Daniel P. W. Ellis. Ground-truth transcriptions of real music from force-aligned MIDI syntheses. *Proceedings of the 4th International Conference on Music Information Retrieval*, pages 135–141, 2003.

[14] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Towards musical query-by-semantic-description using the CAL500 data set. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 439–446. ACM, 2007.