JSYMBOLIC2: EXTRACTING FEATURES FROM SYMBOLIC MUSIC REPRESENTATIONS

Cory McKay
Marianopolis College
cory.mckay@mail.mcgill.ca

Tristano Tenaglia
McGill University
tristano.tenaglia@mail.mcgill.ca

Ichiro Fujinaga
McGill University
ich@music.mcgill.ca

ABSTRACT

This demo presents the jSymbolic2 software for extracting features from symbolic music representations. jSymbolic2 is a tool for assisting musicologists and music theorists in large-scale empirical research projects, and for directly performing the kinds of machine learning based classification and similarity research well-known to the MIR community.

1. INTRODUCTION

A vast amount of significant research performed by music theorists, musicologists, music librarians and music information retrieval researchers in general focuses on symbolic music representations. Unfortunately, relatively few MIR-oriented software tools are available to such researchers for facilitating and carrying out large-scale searches, analyses and empirical studies of the huge quantity of music represented in scores and other symbolic representations.

jSymbolic2 is a free and fully open-source Java tool designed to at least partially address this shortcoming. Its primary function is to easily extract a large number of statistical descriptors / features from digital symbolic music representations. These features can then be used to directly assist researchers in search and analysis-based tasks, or for sophisticated applications using machine-learning. An example of the latter would be to train a model on the styles of different Renaissance composers, and then use this trained model to help determine the composer of pieces whose attribution is uncertain.

jSymbolic2 is a significant expansion of the earlier jSymbolic1 software [6], and is part of the jMIR framework.

2. RELATED RESEARCH

The Humdrum toolkit [4] is quite well-known in the musicological community, and is an excellent tool for analyzing music, although it does not extract features as such. The Melisma Music Analyzer [11] is another quite useful analysis-oriented system.

Relatively few systems designed specifically for extracting features from symbolic music have been published. The MIDI Toolbox [2] is one particularly well-known system implemented in Matlab, and the powerful music21 analysis toolkit incorporates the original jSymbolic1 features [1], and also offers substantial other useful functionality.

Additional work has also been published where symbolic feature extraction is performed as a part of larger research projects. Two interesting standouts, from a feature extraction perspective, are the publications of Ponce de León and Iñesta [9], and of Raffel and Ellis [10].

To the best knowledge of the authors, there is no existing software that extracts anywhere near the number of features as jSymbolic2, nor is there any with the same focus on usability and extensibility.

3. SIMSSA INTEGRATION

One of the key new advantages of jSymbolic2 is its integration with the SIMSSA (Single Interface for Music Score Search and Analysis) project [5,8] and its various components. This project involves applying optical music recognition technology to the huge number of digitized scores held at libraries around the world, and storing the results in easily accessible and searchable databases. Amongst the many other advantages offered by the various SIMSSA components, all of these scores will ultimately automatically have features extracted from them by jSymbolic2.

Not only will this allow music researchers to query scores in relatively traditional ways (e.g. using textual metadata or melodic segments); it will also allow searches based on feature values and ranges. A researcher could thus filter results based on the amount of chromaticism in a piece, for example, or the amount of parallel motion between voices.

All of this functionality will be accessible via the SIMSSA user interfaces, with the technical work done in the background in a distributed and efficient manner using SIMSSA's Rodan workflow management system. Work is currently underway to implement automatic annotation of pieces as they are added to SIMSSA's Elvis database [5,8], with later expansion to the Musiclibs database [3].

4. DATA FORMATS

jSymbolic1 was originally only able to extract features from MIDI files. A significant expansion in jSymbolic2 is the ability to extract features from MEI [12] (and, using a Rodan conversion workflow, MusicXML) files as well.

This is a powerful addition, as it allows myriad types of information that cannot be encoded in MIDI to become accessible for feature extraction. For example, jSymbolic2 can now extract features related to the prevalence of grace notes or of slurs in scores. MEI is also the primary format used in the SIMSSA framework.

jSymbolic2 also incorporates a new custom MEI parser and MEI-to-MIDI converter called jMei2Midi, which can also be used as standalone software. jMei2Midi performs a more extensive level of conversion of MEI than any other known converter, and also maintains a channel for preserving and transmitting information that cannot be represented as MIDI.

5. NEW FEATURES AND EXTENSIBILITY

The first version of the software, jSymbolic1, implemented 111 features, a quite impressive number, but less than the full 160 features proposed [6]. An additional 28 of these proposed features are now implemented in jSymbolic2, and two entirely new features have also been added, for a total of 141 features.

Of particular importance, jSymbolic2 now includes implementations of all of the proposed features associated with vertical intervals and chords, which were absent in jSymbolic1. This now enables harmonically-oriented features to be extracted, which are particularly useful to music scholars. In addition, the 141 current features also include features related to instrumentation, texture, rhythm, dynamics, pitch statistics and melodic intervals.

One of the major advantages of jSymbolic2 is that it is designed to be highly modular and extensible. New features can easily be added by simply extending an existing Java class, and it is easy to incorporate the values of existing features and intermediate representations in new features in order to efficiently iteratively build new features of increasing sophistication. jSymbolic2 automatically handles all infrastructure relating to feature dependencies and scheduling. Substantial unit testing infrastructure has also been added.

All of this makes jSymbolic2 not only a tool for feature extraction, but an effective platform for feature development, testing and implementation.

6. USABILITY AND INTERFACES

Given the diverse types of users for which jSymbolic2 is designed, it is crucial that it be easy to learn and use, and that it be easily adaptable to various use cases. This priority has been essential in informing jSymbolic2's design. jSymbolic2's extensive and detailed new manual also helps to achieve these goals.

The original version of jSymbolic was only usable via a GUI. jSymbolic2 now also includes a flexible command-line interface for batch processing, a clear Java API for programmatic access and a Rodan workflow for distributed use within the SIMSSA framework.

jSymbolic2 also now includes a specialized configuration file that can be used as a way of applying consistent settings across sessions, as well as for keeping a record of the settings used in annotating the contents of musical datasets with extracted feature values.

7. MISCELLANEOUS IMPROVEMENTS

jSymbolic2 now allows windowed feature extraction to be performed, as well as extraction over entire pieces. Although common with audio, this ability to independently extract features from sections of a score is rare in the symbolic domain, and enables powerful kinds of new analysis.

Extracted features can now be saved in the standard Weka ARFF machine learning format. They can also be saved as CSV files, for easy use with a variety of other software. Previously ACE XML was the only output file format option.

There are also numerous other small improvements to jSymbolic2 for which there is insufficient space to describe in detail here.

8. FUTURE WORK

The current priority is to fully annotate both of the SIMSSA databases with extracted jSymbolic features, and to implement an easy-to-use interface for filtering searches based on feature value ranges. Work is also being done on creating infrastructure for automatically extracting features from and annotating new records added to the SIMSSA databases. An additional priority is to also create learned multidimensional combinations of features that allow sophisticated searches (e.g., based on the level of tonality of a piece, where this is estimated based on the values of several existing features).

Work is also continuing on implementing the remaining 21 proposed jSymbolic features, and on working with musicologists and music theorists to design still more entirely new features.

An additional important priority is to carry out machine learning experiments using the jSymbolic2 features. These will include the types of genre classification experiments already conducted with jSymbolic1 [7], both alone and in combination with other types of features (audio, lyrical, cultural, etc.). The main priority, however, will be to work directly with musicologists and music theorists to use jSymbolic2 to help carry out experiments specifically related to the research projects they are working on, with a special focus on work involving very large sets of symbolic data.

9. REFERENCES

[1] Cuthbert, M. S., C. Ariza, and L. Friedland. 2011. Feature extraction and machine learning on symbolic music using the music21 toolkit. *Proceedings of the International Society for Music Information Retrieval Conference.* 387–92.

[2] Eerola, T., and P. Toiviainen. 2004. MIR in Matlab: The MIDI Toolbox. *Proceedings of the International Conference on Music Information Retrieval*. 22–7.

[3] Hankinson, A., and I. Fujinaga. 2016. Cross-institutional music document search. Presented at the *International Association of Music Libraries, Archives and Documentation Centres Congress*.

[4] Huron, D. 2002. Music information processing using the Humdrum toolkit: Concepts, examples, and lessons. *Computer Music Journal* 26 (2): 11–26.

[5] Fujinaga, I., A. Hankinson, and J. Cumming. 2014. Introduction to SIMSSA (Single Interface for Music Score Searching and Analysis). *Proceedings of the International Workshop on Digital Libraries for Musicology*. 100–102.

[6] McKay, C. 2010. Automatic music classification with jMIR. *Ph.D. Dissertation*. McGill University, Canada.

[7] McKay, C., J. A. Burgoyne, J. Hockman, J. B. L. Smith, G. Vigliensoni, and I. Fujinaga. 2010. Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. *Proceedings of the International Society for Music Information Retrieval Conference*. 213–8.

[8] McKay, C., and I. Fujinaga. 2015. Building an infrastructure for a 21st-century global music library. *International Society for Music Information Retrieval Conference Late Breaking and Demo Papers*. Retrieved 29 July 2016, from http://ismir2015.uma.es/LBD/LBD25.pdf.

[9] Ponce de León, P. J., and J. M. Iñesta. 2004. Statistical description models for melody analysis and characterization. *Proceedings of the International Computer Music Conference*. 149–56.

[10] Raffel, C., and D. P. W. Ellis. 2014. Intuitive analysis, creation and manipulation of MIDI data with pretty_midi. *International Society for Music Information Retrieval Conference Late Breaking and Demo Papers*. Retrieved 29 July 2016, from http://www.terasoft.com.tw/conf/ismir2014/LBD/LBD29.pdf.

[11] Temperley, D. 2001. *The cognition of basic musical structures*. Cambridge, MA: MIT Press.

[12] Music Encoding Initiative. Retrieved 29 July 2016, from http://music-encoding.org.