jSymbolic 2: New Developments and Research Opportunities

Cory McKay

Marianopolis College and CIRMMT

Montreal, Canada

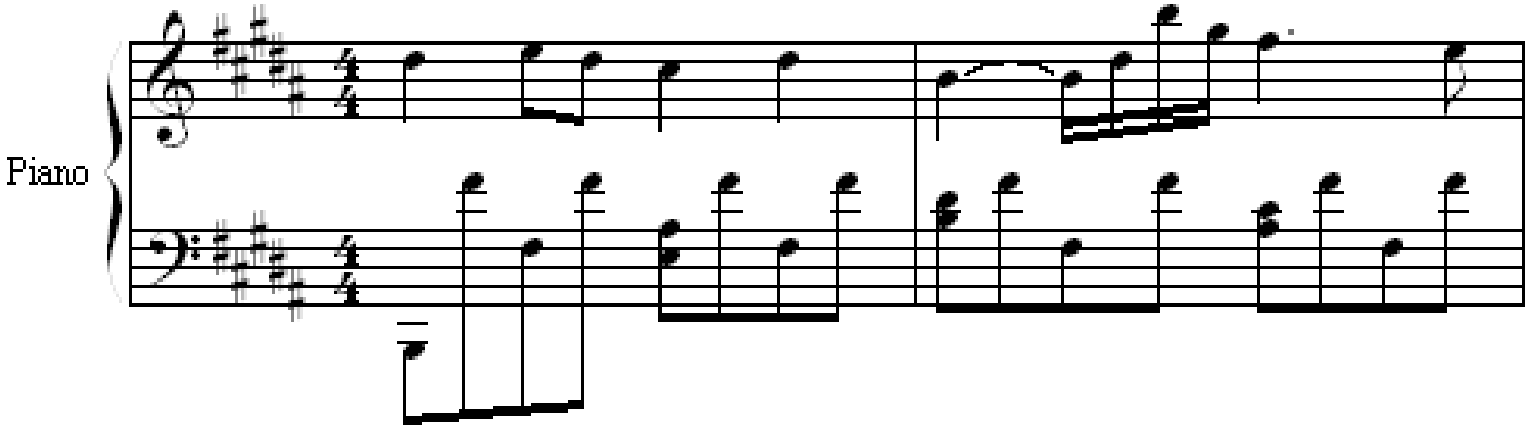
Topics

- Introduction to “features” (from a machine learning perspective)
  - And how they can be useful for musicologists and music theorists
- jSymbolic2
  - What it is
  - How it’s useful to music theorists and musicologists

What are “features”?

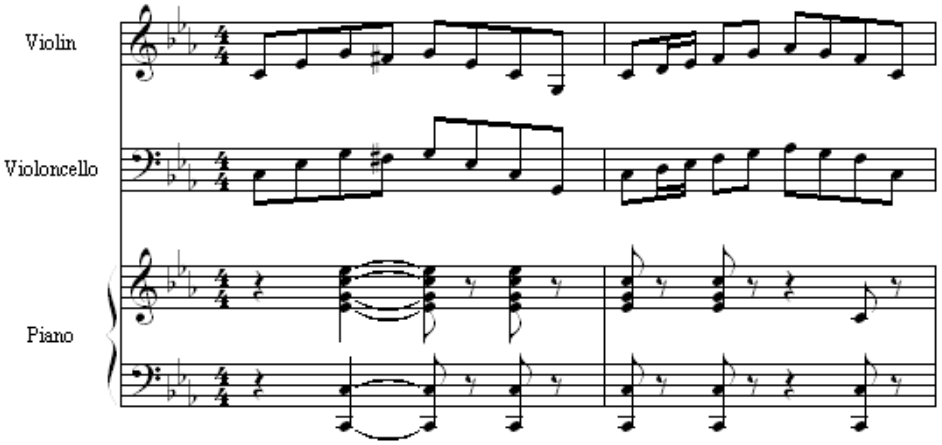
- Pieces of information that can characterize something (e.g. a piece of music) in a (usually) simple way
- (Usually) numerical values
  - Can be single values or can be vectors of related values
  - Histograms are a common type of vector
- (Usually) represent a piece as a whole
  - Or at least regularly spaced windows / musical segments within the piece

Chopin's *Nocturne in B, Op. 32, No. 1*



- Average Note To Note Dynamics Change: 6.03
- Chromatic Motion: 0.0769
- Dominant Spread: 3
- Harmonicity of Two Strongest Rhythmic Pulses: 1
- Importance of Bass Register: 0.2
- Interval Between Strongest Pitch Classes: 3
- Most Common Pitch Class Prevalence: 0.433
- Note Density: 3.75
- Number of Common Melodic Intervals: 3
- Number of Strong Pulses: 5
- Orchestral Strings Fraction: 0
- Overall Dynamic Range: 62
- Pitch Class Variety: 7
- Range: 48
- Relative Strength of Most Common Intervals: 0.5
- Size of Melodic Arcs: 11
- Stepwise Motion: 0.231
- Strength of Strongest Rhythmic Pulse: 0.321
- Variability of Note Duration: 0.293
- Variation of Dynamics: 16.4

Mendelssohn's *Piano Trio No. 2*

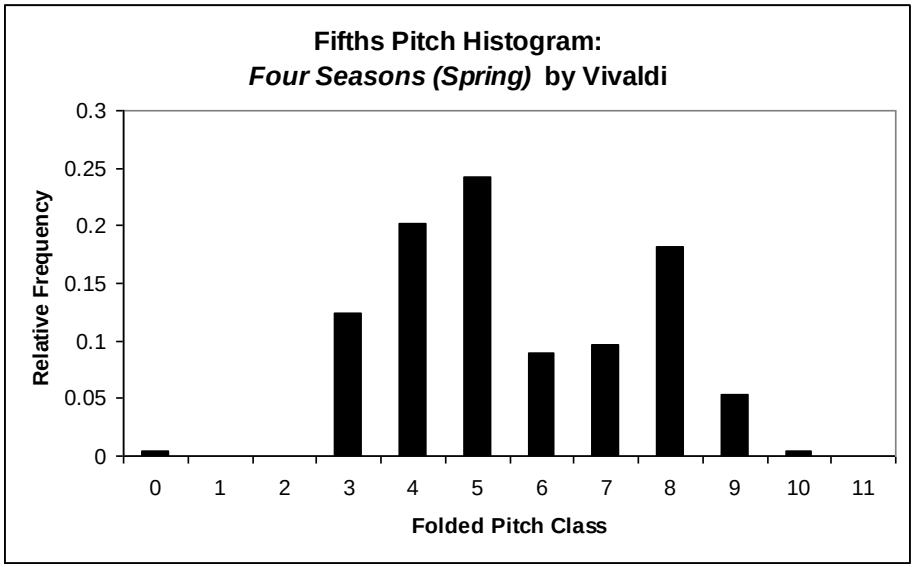


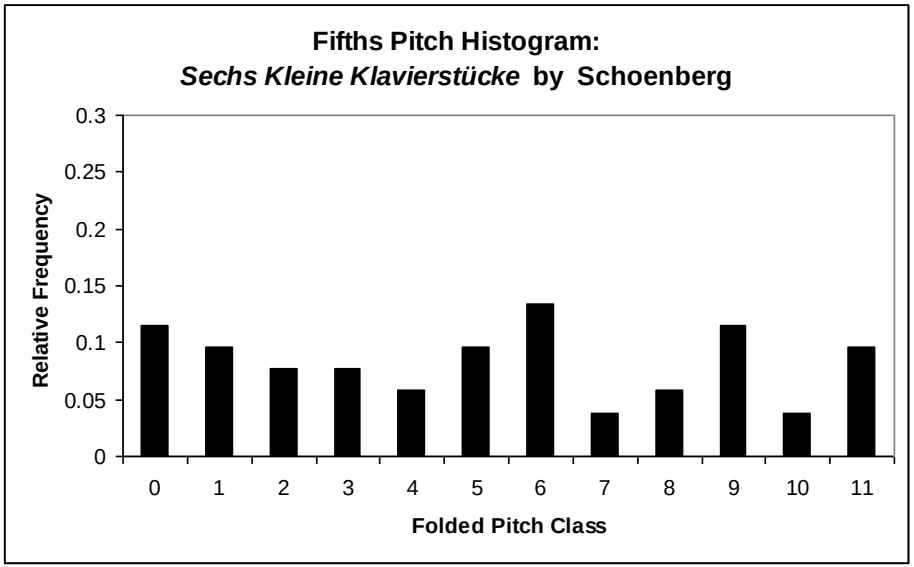
- Average Note To Note Dynamics Change: 1.46
- Chromatic Motion: 0.244
- Dominant Spread: 2
- Harmonicity of Two Strongest Rhythmic Pulses: 1
- Importance of Bass Register: 0.373
- Interval Between Strongest Pitch Classes: 7
- Most Common Pitch Class Prevalence: 0.39
- Note Density: 29.5
- Number of Common Melodic Intervals: 6
- Number of Strong Pulses: 6
- Orchestral Strings Fraction: 0.56
- Overall Dynamic Range: 22
- Pitch Class Variety: 7
- Range: 39
- Relative Strength of Most Common Intervals: 0.8
- Size of Melodic Arcs: 7.27
- Stepwise Motion: 0.439
- Strength of Strongest Rhythmic Pulse: 0.173
- Variability of Note Duration: 0.104
- Variation of Dynamics: 5.98

Feature value comparison

Feature	*Nocturne*	*Trio*
Average Note To Note Dynamic Change	6.03	1.46
Overall Dynamic Range	62	22
Variation of Dynamics	16.4	5.98
Note Density	3.75	29.5
Orchestral Strings Fraction	0	**0.56**
Variability of Note Duration	**0.293**	0.104
Chromatic Motion	0.077	**0.244**
Range	48	39

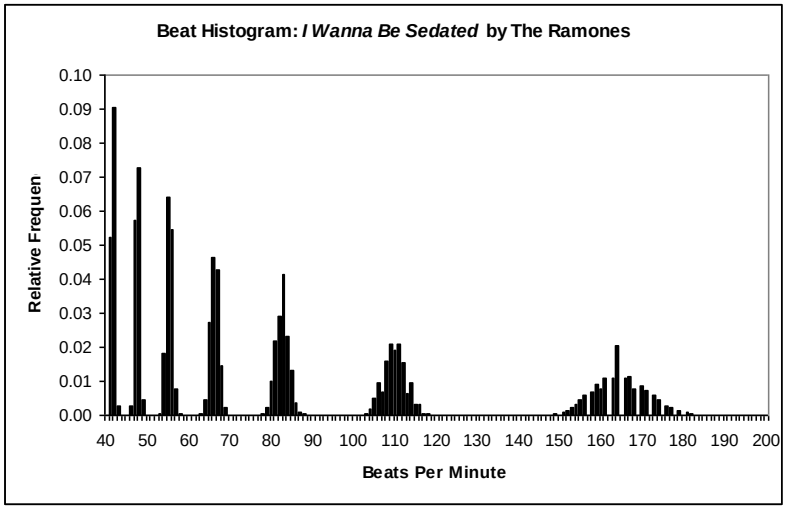
Fifths pitch class histogram

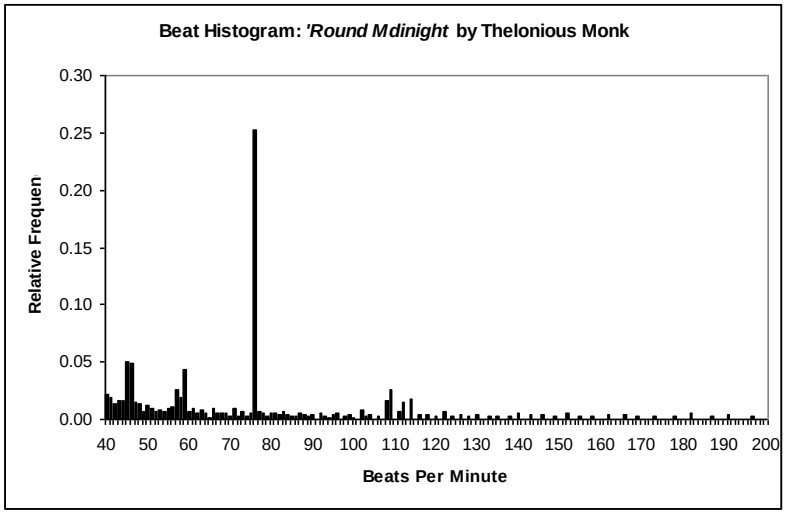




Beat histogram

- Beat histograms use a technique called "autocorrelation" to calculate the relative strengths of different beat periodicities
- "I Wanna Be Sedated" by The Ramones (top)
  - Several harmonic peaks with large spreads around them
- "'Round Midnight" by Thelonious Monk (bottom)
  - Only one strong peak, with a large low-level spread
- Histograms like this can be used directly, or other features may be derived from them
  - e.g. peak statistics





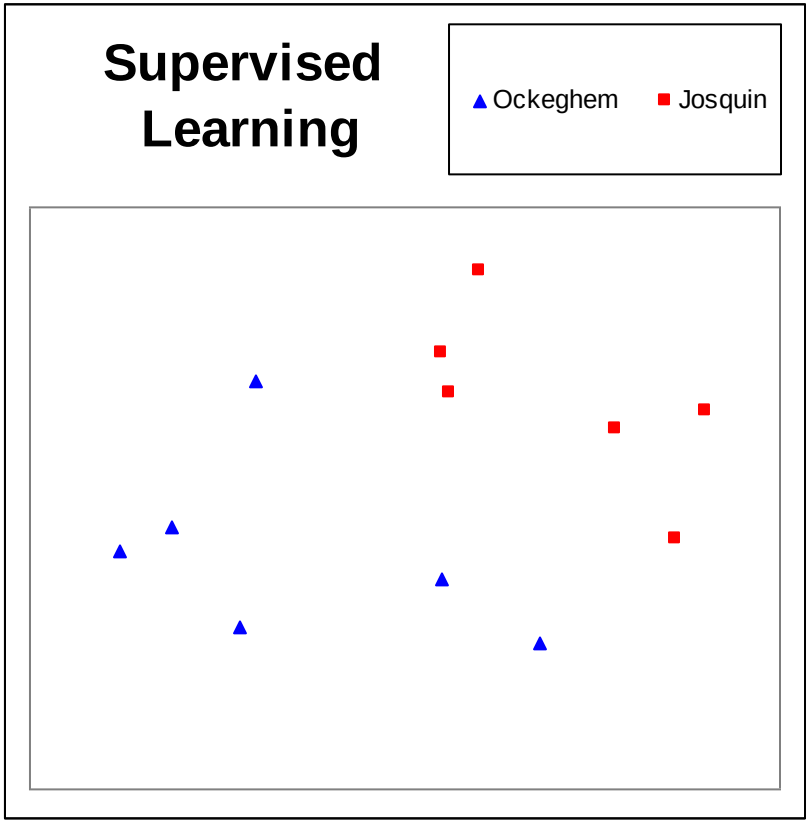
How can features be useful?

- Sophisticated searches of large musical databases
  - e.g. find all pieces with no more than X amount of chromaticism, and less than Y amount of parallel motion
  - ELVIS database + Musiclibs
- Using statistical analysis and visualization tools to study the empirical musical importance of various features when extracted from large datasets
  - e.g. features based on instrumentation were most effective for distinguishing genres (McKay & Fujinaga 2005)
- Using machine learning to classify or cluster music
  - Supervised or unsupervised learning
  - e.g. identify the composers of unattributed musical pieces

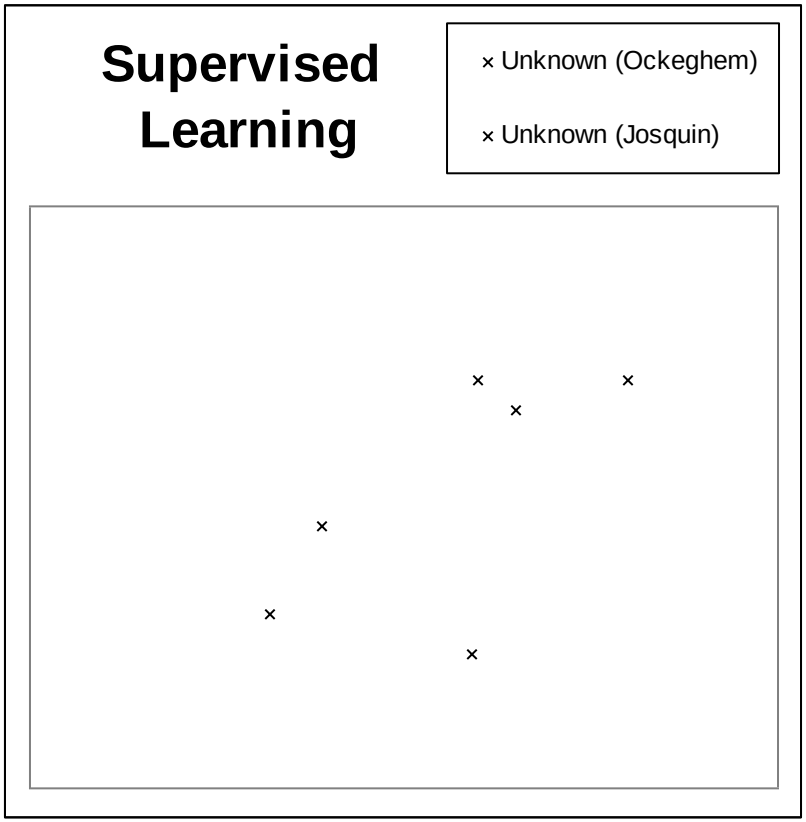
Sample expert system

```
if ( parallel_fifths == 0 &&
     landini_cadences == 0 )
   then composer → Palestrina
else composer → Machaut
```

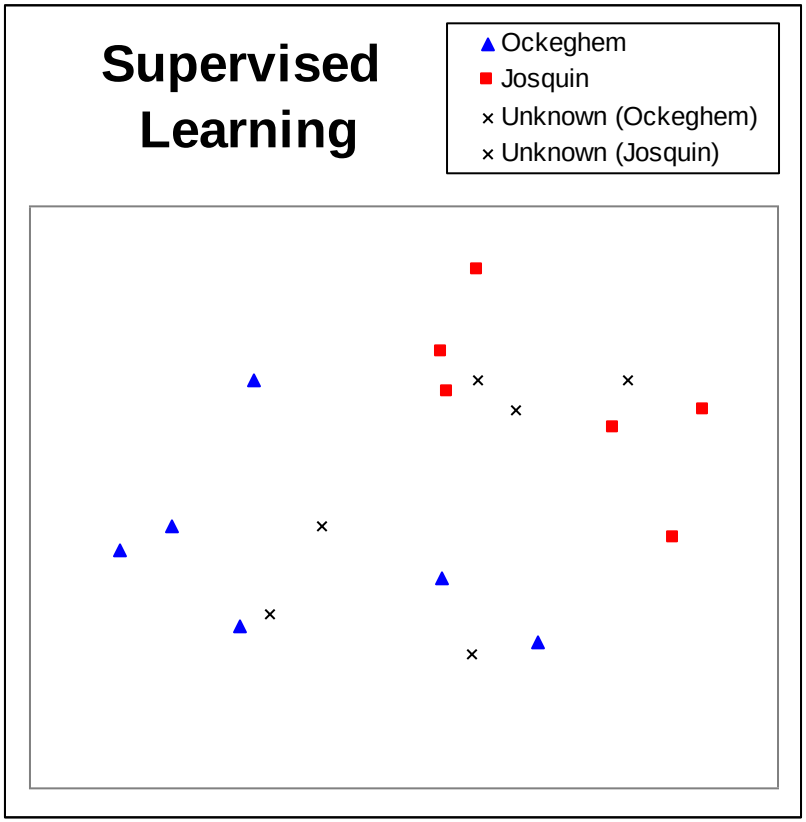
Sample supervised learning



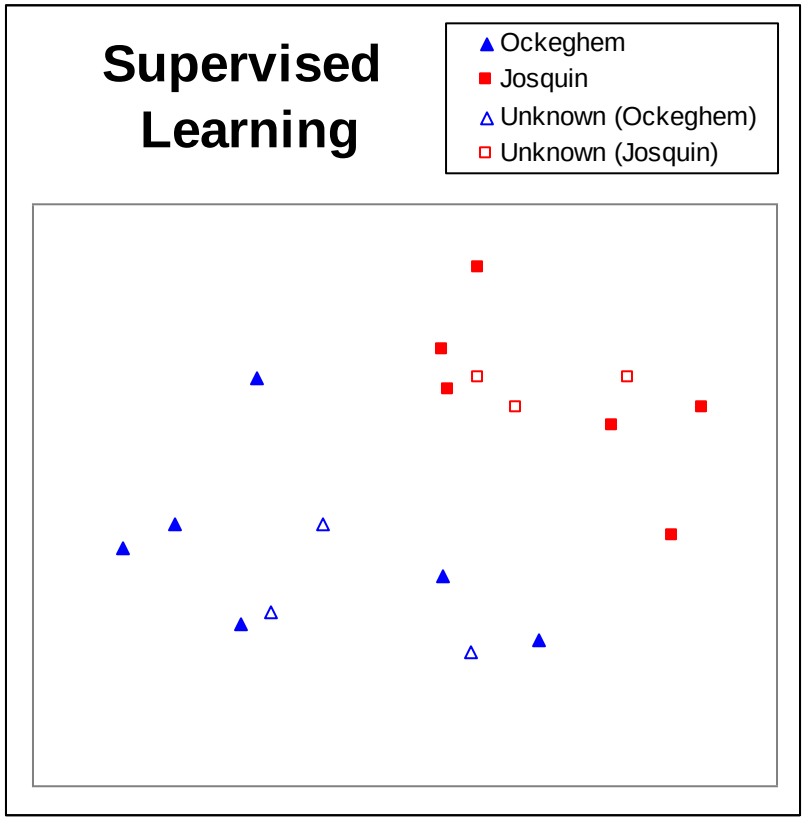
Sample supervised learning



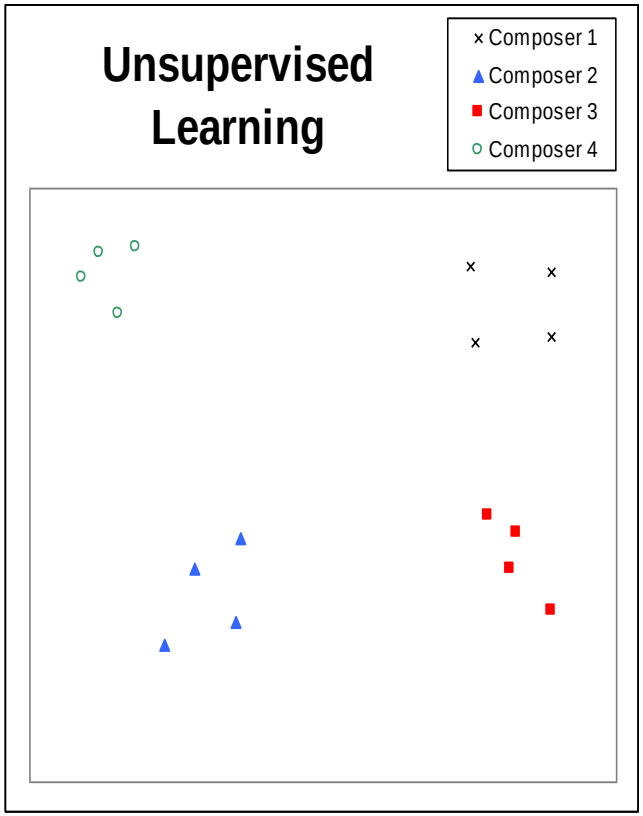
Sample supervised learning



Sample supervised learning



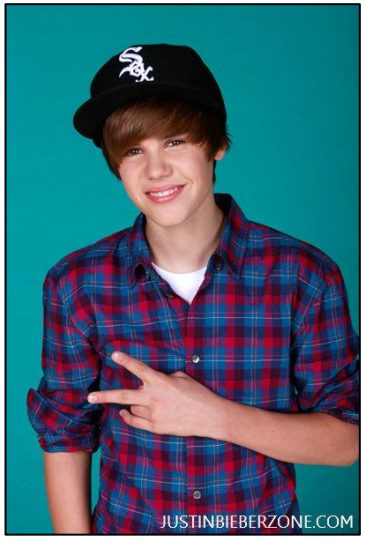
Sample unsupervised learning



Benefits of features and machine learning

- Can quickly perform consistent empirical studies involving thousands of pieces
- Can be applied to diverse types of music
- Can simultaneously consider thousands of features and their interrelationships
  - And can statistical condense many features into low-dimensional spaces when needed
- No need to formally specify any heuristics or queries before beginning analyses
  - Unless you want to, of course
- Can avoid (or validate) potentially incorrect ingrained biases and assumptions

jSymbolic's lineage



- Bodhidharma (2004)
  - Specialized feature extraction and machine learning for genre classification research
- jSymbolic (2006)
  - General-purpose feature extraction
  - Part of jMIR
- jSymbolic2 (2016)
  - Bigger and better!

What does jSymbolic2 do?

- Extracts 158 features
- Some of these are multi-dimensional histograms, including:
  - Pitch and pitch class histograms
  - Melodic interval histogram
  - Vertical interval histograms
  - Chord types histogram
  - Beat histogram
  - Instrument histograms

jSymbolic2's feature types (1/2)

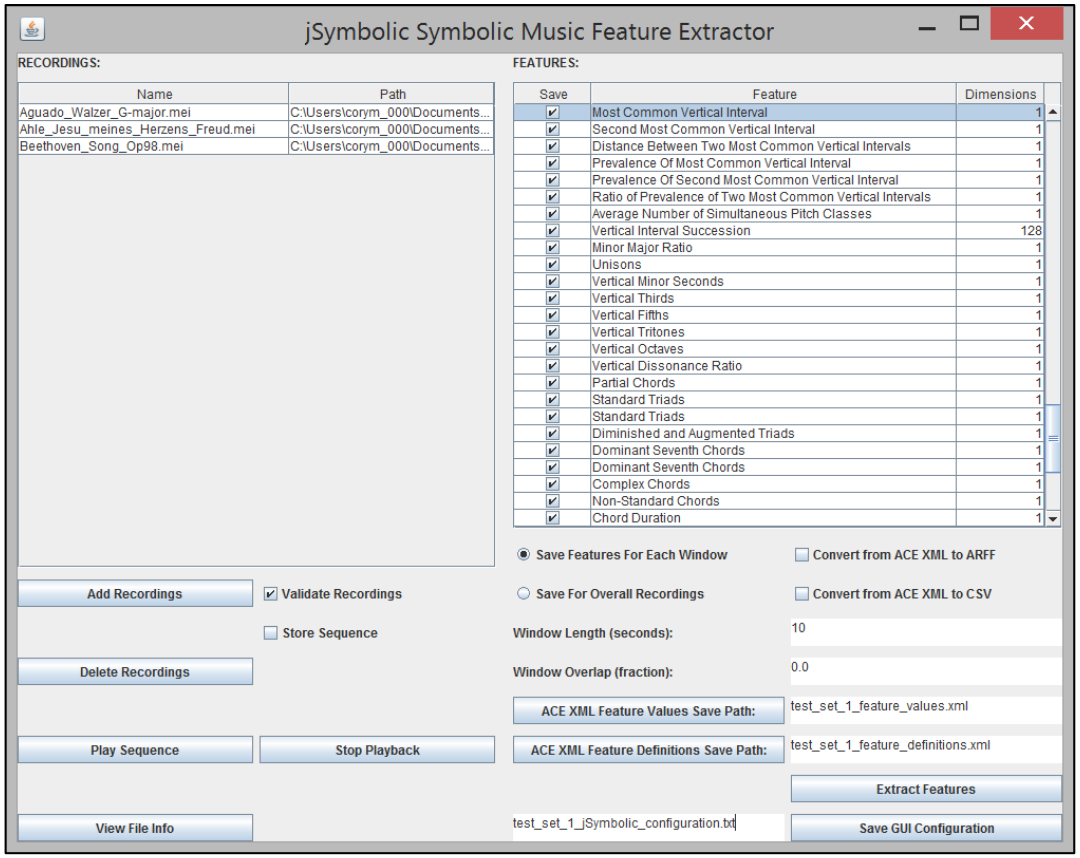
- Instrumentation:
  - What types of instruments are present and which are given particular importance relative to others?
- Texture:
  - How many independent voices are there and how do they interact (e.g., polyphonic, homophonic, etc.)?
- Rhythm:
  - Time intervals between the attacks of different notes
  - Duration of notes
  - What kinds of meters and rhythmic patterns are present?
  - Rubato?
- Dynamics:
  - How loud are notes and what kinds of dynamic variations occur?

jSymbolic feature types (2/2)

- Pitch Statistics:
  - What are the occurrence rates of different pitches and pitch classes?
  - How tonal is the piece?
  - How much variety in pitch is there?
- Melody:
  - What kinds of melodic intervals are present?
  - How much melodic variation is there?
  - What kinds of melodic contours are used?
  - What types of phrases are used?
- Chords:
  - What vertical intervals are present?
  - What types of chords do they represent?
  - How much harmonic movement is there?

How can you use jSymbolic2

- Graphical user interface
- Command line interface
- Rodan workflow
- Java API

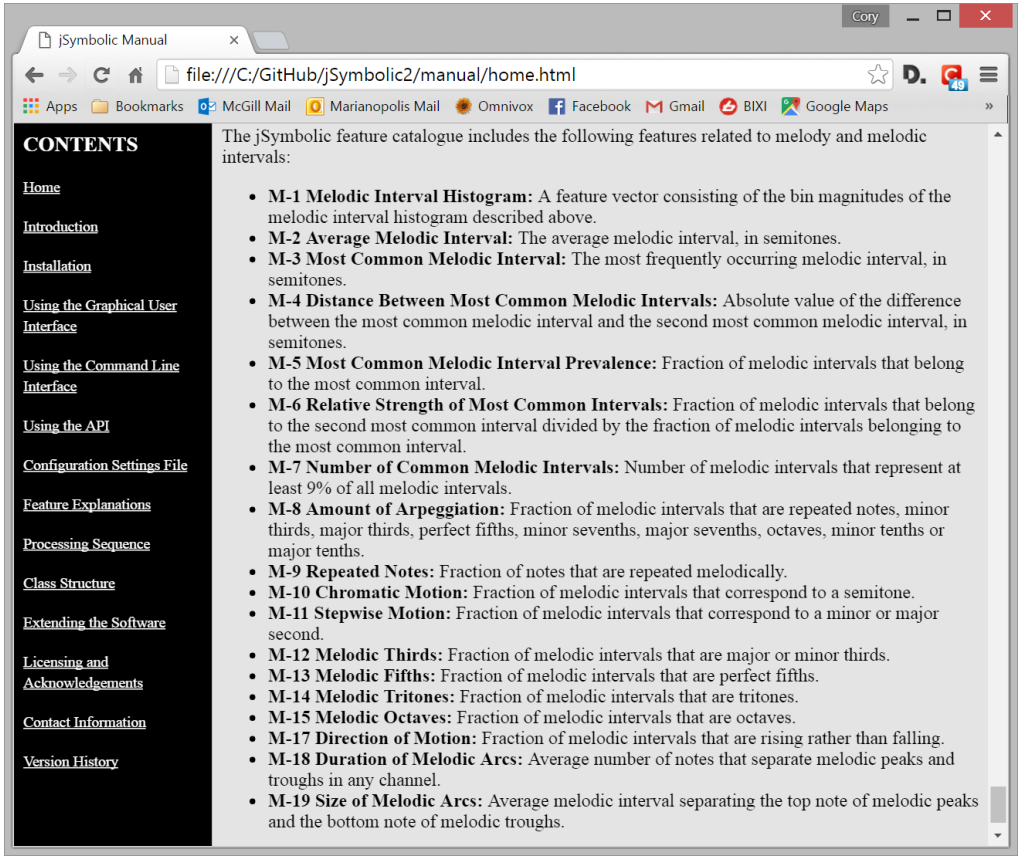


jSymbolic2's file formats

- Input:
  - MIDI
  - MEI
  - MusicXML (via Rodan workflow only)
- Output:
  - ACE XML
  - Weka ARFF
  - CSV

jSymbolic2's documentation

- Super-mega-ultra detailed manual
  - At least compared to most academic software manuals
  - In HTML
- Super-mega-ultra detailed Javadocs
  - For programmers



jSymbolic2: More great things

- Windowed feature extraction
  - Including overlapping windows
- Configuration files
  - Pre-set feature choices
  - Pre-set input and output choices
  - More
- jMei2Midi
  - Most complete MEI to MIDI converter in the universe!
  - General-purpose (not just for jSymbolic2)
  - Specialized pipeline for transmitting relevant MEI data that cannot be represented in MIDI

Exploratory simple pilot study

- Josquin vs. Ockeghem composer identification / attribution
  - 124 jSymbolic2 features extracted from the JRP data
  - 105 Josquin pieces and 98 Ockeghem
- Achieved 89.7% classification accuracy
  - 10-fold cross-validation
- Lots of room for improving results still further
  - Only used simple SVM classifier with default settings
  - No dimensionality reduction was used
    - Both expert insights and automatic analysis can be applied
  - Still more jSymbolic2 features to come
- Interesting future research applications:
  - Determine which features are most effective
    - Can analyze feature data both visually and statistically
  - Apply trained classifiers to unattributed or uncertain pieces
  - Expand scope to other composers

What you can do with jSymbolic

- Empirically study huge collections of music in new ways
  - Search music databases based on feature values
  - Analyze and visualize music based on feature values
  - Use machine learning
- Design your own custom features
  - jSymbolic2 is specifically designed to make it easy to add new custom features
  - Easy to iteratively build increasingly complex features based on existing features
- Perform multimodal research
  - Combine symbolic features with other features extracted from audio, lyrics and cultural data
  - This improves results substantially! (McKay et al. 2010)

Use jSymbolic2 with jMIR

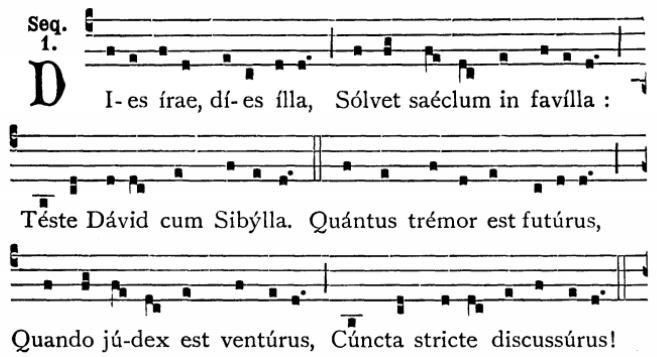
- ACE: Meta-learning classification engine
- Bodhidharma MIDI, SLAC and Codaich: datasets
- jAudio: Audio feature extraction
- jLyrics: Extracts features from lyrical transcriptions
- jWebMiner: Cultural feature extraction
- ACE XML: File formats
  - Features, feature metadata, instance metadata and ontologies
- lyricFetcher: Lyric mining
- jMusicMetaManager: Metadata management
- jSongMiner: Metadata harvesting
- jMIRUtilities: Infrastructure for conducting experiments
- jProductionCritic: Automated production error-checking

Research collaborations

- We would love to collaborate with music theorists and musicologists on their work
- We can help you apply and adapt jSymbolic to specific research projects
- We can help you come up with novel ways to study music

jSymbolic2: Currently in progress

- Final testing and debugging
- Annotation of all valid files in the ELVIS database with extracted features
  - And Musiclibs, eventually
  - Auto-annotation scripts
- MEI pre-modern notation
- Designing new features
  - Requests welcome!



Acknowledgements

- Tristano Tenaglia
  - Implemented almost all of the new jSymbolic2 code
- Véronique Lagacé
  - ELVIS database integration scripts
- Ryan Bannon, Dr. Andrew Hankinson and Dr. Reiner Krämer
  - On-site Rodan and ELVIS expertise in the lab
- Prof. Ichiro Fujinaga and Prof. Julie Cumming
  - Grant application and project supervision superstardom
- The FRQSC and SSHRC
  - Great financial generosity

Thanks for your attention

- **E-mail:** cory.mckay@mail.mcgill.ca
- **jSymbolic2:** github.com/DDMAL/jSymbolic2
- **jMIR:** jmir.sourceforge.net

Schulich School of Music
École de musique Schulich

Centre for Interdisciplinary Research in Music Media and Technology

Social Sciences and Humanities Research Council of Canada

Conseil de recherches en sciences humaines du Canada

DISTRIBUTED DIGITAL MUSIC ARCHIVES & LIBRARIES LAB