jMIR Overview

Cory McKay

Marianopolis College and CIRMMT

Montreal, Canada

Lecture contents

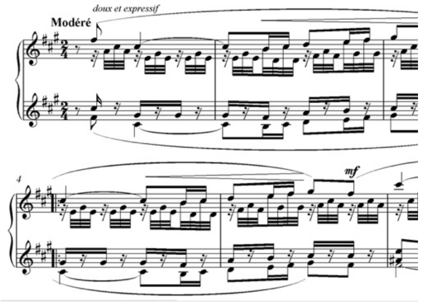
- Introduction to automatic music classification
- Introduction to the jMIR software
- The jMIR components
  - One by one
- Paper ideas
  - Previous collaborations

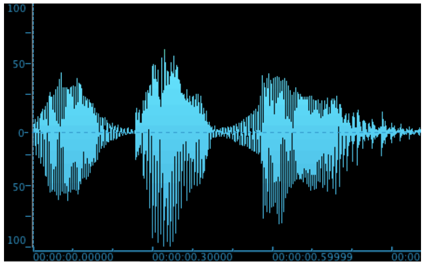
Goal of automatic music classification

- Learn some way of mapping “features” extracted from an “instance” to one or more “classes”
  - Instance: an item to be classified
    - e.g. a song
  - Features: representative information extracted from an instance
    - e.g. amount or chromatic motion in a song
  - Class: a category of interest
    - e.g. a genre, mood, artist, composer, instrument, etc.
    - Ideally organized into some kind of class ontology
- This mapping is typically learned using some form of pattern recognition and machine learning

Main sources of information

- Symbolic recordings
  - e.g. MIDI
- Audio recordings
  - e.g. MP3
- Cultural data
  - e.g. web data, listener statistics, metadata tags, etc.
- Lyrics
- Others
  - Album art, videos, etc.





Automatic music classification

- Typical procedure:
  - Collect annotated training / testing data
  - Extract features
  - Reduce feature dimensionality
  - Train a classification model
    - Typically supervised
  - Validate the model
- Some significant challenges:
  - Acquiring a sufficiently large dataset with sufficiently high-quality annotations
  - Designing features that encapsulate relevant data

Overview of the jMIR software

- jMIR is software suite designed for performing research in automatic music classification
- Primary tasks performed:
  - Dataset management
    - Acquiring, correcting and organizing metadata
  - Feature extraction
  - Machine learning
  - Data storage file formats

Some characteristics of jMIR

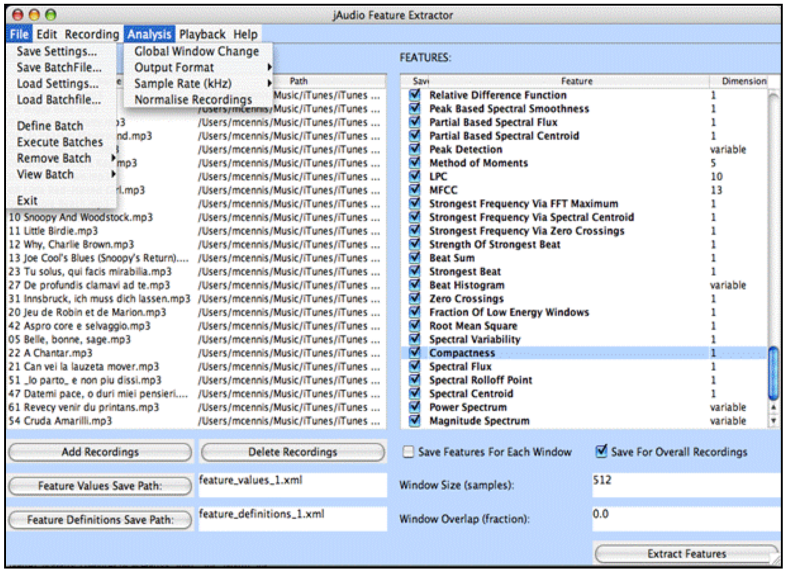
- Has a separate software component to address each important aspect of automatic music classification
  - Each component can be used independently
  - jMIR can also be used as an integrated whole
- Free and open source
- Architectural emphasis on providing an extensible platform for iteratively developing new techniques and algorithms
- Interfaces designed for both technical and non-technical users
- Facilitates multimodal research

jMIR components

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

jAudio: Audio feature extractor

- Extracts features from audio files
  - MP3, WAV, AIFF, AU, SND, etc.
- 28 bundled core features
  - Mainly low-level, some high-level
- Can automatically generate new features using metafeatures and aggregators
  - e.g. the change in a feature value from window to window
- Includes tools for testing new features being developed
  - Synthesize audio, record audio, sonify MIDI, display audio, etc.



e	Path
	Music/iTunes/iTunes ...
	/users/mcennis/Music/iTunes/iTunes ...
)3	/Users/mcennis/Music/iTunes/iTunes ...
nd.mp3	/Users/mcennis/Music/iTunes/iTunes ...
3	/Users/mcennis/Music/iTunes/iTunes ...
mp3	/Users/mcennis/Music/iTunes/iTunes ...
	/Users/mcennis/Music/iTunes/iTunes ...
rl.mp3	/Users/mcennis/Music/iTunes/iTunes ...
	/Users/mcennis/Music/iTunes/iTunes ...
10 Snoopy And Woodstock.mp3	/Users/mcennis/Music/iTunes/iTunes ...
11 Little Birdie.mp3	/Users/mcennis/Music/iTunes/iTunes ...
12 Why, Charlie Brown.mp3	/Users/mcennis/Music/iTunes/iTunes ...
13 Joe Cool's Blues (Snoopy's Return)....	/Users/mcennis/Music/iTunes/iTunes ...
23 Tu solus, qui facis mirabilia.mp3	/Users/mcennis/Music/iTunes/iTunes ...
27 De profundis clamavi ad te.mp3	/Users/mcennis/Music/iTunes/iTunes ...
31 Innsbruck, ich muss dich lassen.mp3	/Users/mcennis/Music/iTunes/iTunes ...
20 Jeu de Robin et de Marion.mp3	/Users/mcennis/Music/iTunes/iTunes ...
42 Aspro core e selvaggio.mp3	/Users/mcennis/Music/iTunes/iTunes ...
05 Belle, bonne, sage.mp3	/Users/mcennis/Music/iTunes/iTunes ...
22 A Chantar.mp3	/Users/mcennis/Music/iTunes/iTunes ...
21 Can vei la lauzeta mover.mp3	/Users/mcennis/Music/iTunes/iTunes ...
51 _lo parto_ e non piu dissi.mp3	/Users/mcennis/Music/iTunes/iTunes ...
47 Datemi pace, o duri miei pensieri....	/Users/mcennis/Music/iTunes/iTunes ...
61 Revecy venir du printans.mp3	/Users/mcennis/Music/iTunes/iTunes ...
54 Cruda Amarilli.mp3	/Users/mcennis/Music/iTunes/iTunes ...

FEATURES:

Savi	Feature	Dimension
☑	Relative Difference Function	1
☑	Peak Based Spectral Smoothness	1
☑	Partial Based Spectral Flux	1
☑	Partial Based Spectral Centroid	1
☑	Peak Detection	variable
☑	Method of Moments	5
☑	LPC	10
☑	MFCC	13
☑	Strongest Frequency Via FFT Maximum	1
☑	Strongest Frequency Via Spectral Centroid	1
☑	Strongest Frequency Via Zero Crossings	1
☑	Strength Of Strongest Beat	1
☑	Beat Sum	1
☑	Strongest Beat	1
☑	Beat Histogram	variable
☑	Zero Crossings	1
☑	Fraction Of Low Energy Windows	1
☑	Root Mean Square	1
☑	Spectral Variability	1
☑	Compactness	1
☑	Spectral Flux	1
☑	Spectral Rolloff Point	1
☑	Spectral Centroid	1
☑	Power Spectrum	variable
☑	Magnitude Spectrum	variable

☐ Save Features For Each Window ☑ Save For Overall Recordings

Feature Values Save Path: feature_values_1.xml

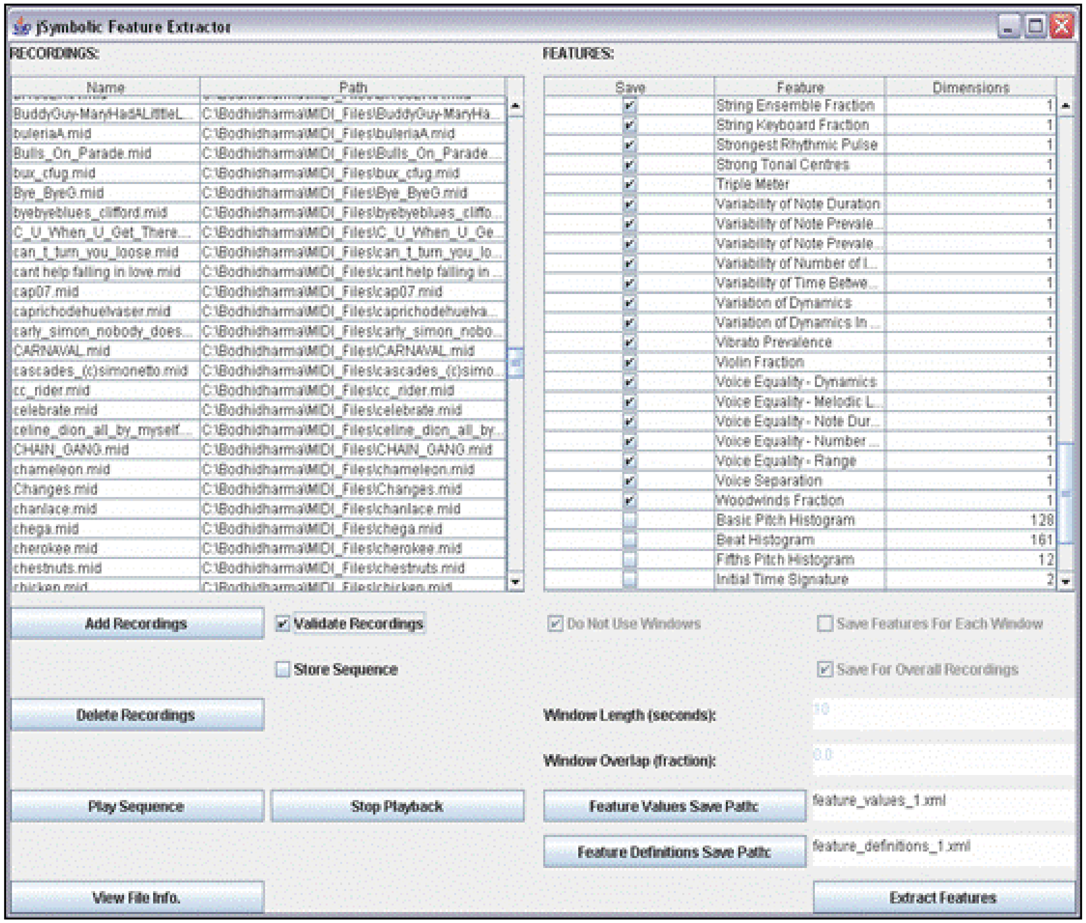
Feature Definitions Save Path: feature_definitions_1.xml

Window Size (samples): 512

Window Overlap (fraction): 0.0

jSymbolic: Symbolic feature extractor

- Extracts features from MIDI files
- 111 implemented features
  - By far the largest existing symbolic feature catalogue
  - Many are original
- An additional 49 features are proposed but not yet implemented



jSymbolic Feature Extractor

RECORDINGS:

Name	Path
BuddyGuy-MaryHadALittleL...	C:\Bodhidharma\MIDI_Files\BuddyGuy-MaryHa...
buleriaA.mid	C:\Bodhidharma\MIDI_Files\buleriaA.mid
Bulls_On_Parade.mid	C:\Bodhidharma\MIDI_Files\Bulls_On_Parade...
bux_cfug.mid	C:\Bodhidharma\MIDI_Files\bux_cfug.mid
Bye_ByeG.mid	C:\Bodhidharma\MIDI_Files\Bye_ByeG.mid
byebyeblues_clifford.mid	C:\Bodhidharma\MIDI_Files\byebyeblues_cliffo...
C_U_When_U_Get_There...	C:\Bodhidharma\MIDI_Files\C_U_When_U_Ge...
can_t_turn_you_loose.mid	C:\Bodhidharma\MIDI_Files\can_t_turn_you_lo...
cant help falling in love.mid	C:\Bodhidharma\MIDI_Files\cant help falling in ...
cap07.mid	C:\Bodhidharma\MIDI_Files\cap07.mid
caprichodehuelvaser.mid	C:\Bodhidharma\MIDI_Files\caprichodehuelva...
carly_simon_nobody_does...	C:\Bodhidharma\MIDI_Files\carly_simon_nobo...
CARNAVAL.mid	C:\Bodhidharma\MIDI_Files\CARNAVAL.mid
cascades_(c)simonetto.mid	C:\Bodhidharma\MIDI_Files\cascades_(c)simo...
cc_rider.mid	C:\Bodhidharma\MIDI_Files\cc_rider.mid
celebrate.mid	C:\Bodhidharma\MIDI_Files\celebrate.mid
celine_dion_all_by_myself...	C:\Bodhidharma\MIDI_Files\celine_dion_all_by...
CHAIN_GANG.mid	C:\Bodhidharma\MIDI_Files\CHAIN_GANG.mid
chameleon.mid	C:\Bodhidharma\MIDI_Files\chameleon.mid
Changes.mid	C:\Bodhidharma\MIDI_Files\Changes.mid
chanlace.mid	C:\Bodhidharma\MIDI_Files\chanlace.mid
chega.mid	C:\Bodhidharma\MIDI_Files\chega.mid
cherokee.mid	C:\Bodhidharma\MIDI_Files\cherokee.mid
chestnuts.mid	C:\Bodhidharma\MIDI_Files\chestnuts.mid
chicken.mid	C:\Bodhidharma\MIDI_Files\chicken.mid

FEATURES:

Save	Feature	Dimensions
☑	String Ensemble Fraction	1
☑	String Keyboard Fraction	1
☑	Strongest Rhythmic Pulse	1
☑	Strong Tonal Centres	1
☑	Triple Meter	1
☑	Variability of Note Duration	1
☑	Variability of Note Prevale...	1
☑	Variability of Note Prevale...	1
☑	Variability of Number of I...	1
☑	Variability of Time Betwe...	1
☑	Variation of Dynamics	1
☑	Variation of Dynamics In ...	1
☑	Vibrato Prevalence	1
☑	Violin Fraction	1
☑	Voice Equality - Dynamics	1
☑	Voice Equality - Melodic L...	1
☑	Voice Equality - Note Dur...	1
☑	Voice Equality - Number ...	1
☑	Voice Equality - Range	1
☑	Voice Separation	1
☑	Woodwinds Fraction	1
□	Basic Pitch Histogram	128
□	Beat Histogram	161
□	Fifths Pitch Histogram	12
□	Initial Time Signature	2

☑ Validate Recordings

□ Store Sequence

☑ Do Not Use Windows

□ Save Features For Each Window

☑ Save For Overall Recordings

Window Length (seconds): 10

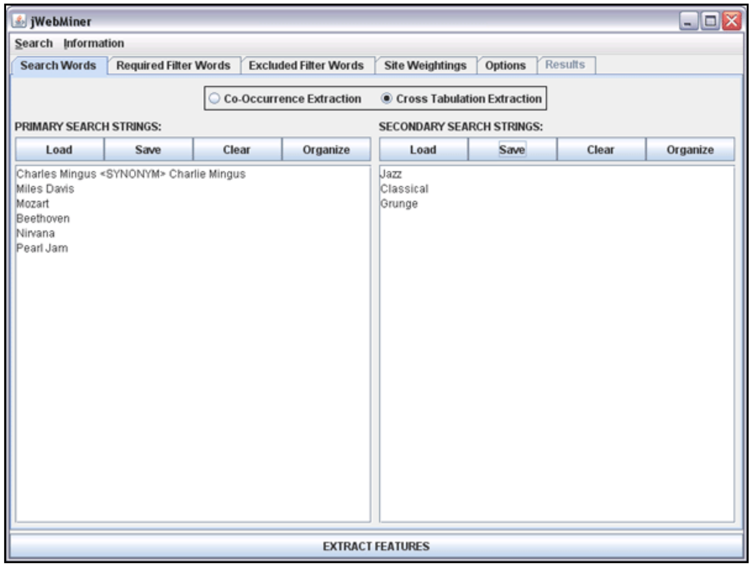
Window Overlap (fraction): 0.0

Feature Values Save Path: feature_values_1.xml

Feature Definitions Save Path: feature_definitions_1.xml

jWebMiner: Cultural feature extractor

- Extracts cultural features from the web using search engine web services
- Calculates how often particular strings co-occur on the same web pages
  - e.g. how often does “J. S. Bach” co-occur on a web page with “Baroque”, compared to “Prokofiev”?
  - Results are processed to remove noise
- Additional options:
  - Can assign weights to particular sites
  - Can enforce filter words
  - Permits synonyms
- Also calculates features based on Last.FM user tags frequencies



○ Co-Occurrence Extraction ● Cross Tabulation Extraction

PRIMARY SEARCH STRINGS:

Charles Mingus <SYNONYM> Charlie Mingus
Miles Davis
Mozart
Beethoven
Nirvana
Pearl Jam

SECONDARY SEARCH STRINGS:

Jazz
Classical
Grunge

lyricFetcher: Lyric miner

- lyricFetcher automatically harvests lyrics from on-line lyrics repositories
    - LyricWiki and LyricsFly
    - Queries based on lists of song titles and artist names
- Post-processing is applied to the lyrics in order to remove noise and make them sufficiently consistent for feature extraction
    - Deals with situations where sections of lyrics are abridged using keywords such as “chorus”, “bridge”, “verse”, etc.
    - Filters out keywords that could contaminate the lyrics

jLyrics: Lyrical feature extractor

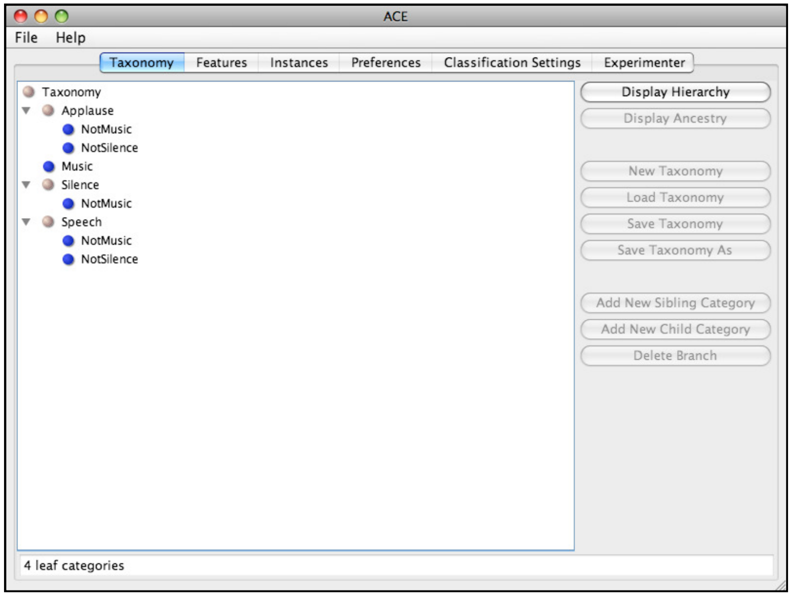
- Extracts features from lyrics stored in text files:
  - Automated Readability Index
  - Average Syllable Count Per Word
  - Contains Words
  - Flesh-Kincaid Grade Level
  - Flesh Reading Ease
  - Function Word Frequencies
  - Letter-Bigram Components
  - Letter Frequencies
  - Letters Per Word Average
  - Letters Per Word Variance
  - Lines Per Segment Average
  - Lines Per Segment Variance
  - Number of Lines

  Number of Segments
  Number of Words
  Part-of-Speech Frequencies
  Punctuation Frequencies
  Rate of Misspelling
  Sentence Count
  Sentence Length Average
  Topic Membership Probabilities
  Vocabulary Richness
  Vocabulary Size
  Word Profile Match
  Words Per Line Average
  Words Per Line Variance

- Can also automatically generate word frequency profiles for particular classes of training data is provided

ACE: Meta-learning engine

- Evaluates the relative suitability of different dimensionality reduction and classification algorithms for a given problem
  - Can also train and classify with manually selected algorithms
- Evaluates algorithms in terms of
  - Classification accuracy
  - Consistency
  - Time complexity
- Based on the Weka framework, so new algorithms can be added easily

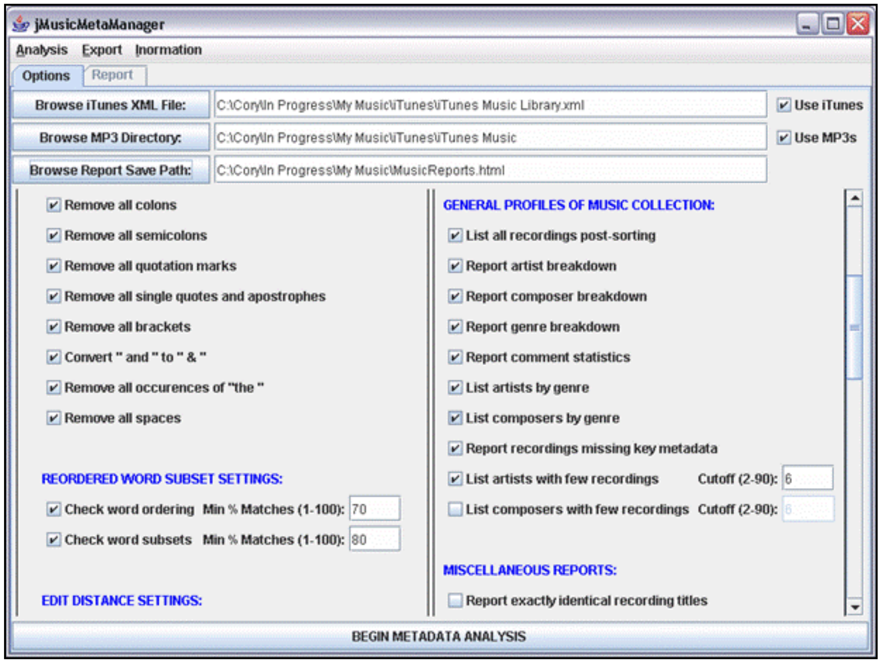


jMIR datasets

- Codaich is an MP3 research set
  - Carefully cleaned and labelled
  - The published 2006 version has 26,420 recordings
    - Belonging to 55 genres
    - Is constantly growing: currently over 45,000 MP3s
- Bodhidharma MIDI has 950 MIDI recordings
  - 38 genres of music
- SLAC consists of 250 matched audio recordings, MIDI recordings, lyrical transcriptions and metadata that can be used to extract cultural features
  - Useful for experiments on combining features from different types of data
  - 10 genres of music (in 5 pairs of similar genres)

jMusicMetaManager: Dataset manager

- Detects metadata errors/inconsistencies and redundant copies of recordings
- Detects differing metadata values that should in fact be the same
  - e.g. “Charlie Mingus” vs. “Mingus, Charles”
- Generates HTML inventory and profile reports
  - 39 reports in all
- Parses metadata from ID3 tags and iTunes XML



Browse iTunes XML File: C:\Cory\In Progress\My Music\iTunes\iTunes Music Library.xml ☑ Use iTunes

Browse MP3 Directory: C:\Cory\In Progress\My Music\iTunes\iTunes Music ☑ Use MP3s

Browse Report Save Path: C:\Cory\In Progress\My Music\MusicReports.html

☑ Remove all colons
☑ Remove all semicolons
☑ Remove all quotation marks
☑ Remove all single quotes and apostrophes
☑ Remove all brackets
☑ Convert " and " to " & "
☑ Remove all occurences of "the "
☑ Remove all spaces

REORDERED WORD SUBSET SETTINGS:

☑ Check word ordering Min % Matches (1-100): 70
☑ Check word subsets Min % Matches (1-100): 80

EDIT DISTANCE SETTINGS:

GENERAL PROFILES OF MUSIC COLLECTION:

☑ List all recordings post-sorting
☑ Report artist breakdown
☑ Report composer breakdown
☑ Report genre breakdown
☑ Report comment statistics
☑ List artists by genre
☑ List composers by genre
☑ Report recordings missing key metadata
☑ List artists with few recordings Cutoff (2-90): 6
□ List composers with few recordings Cutoff (2-90): 6

MISCELLANEOUS REPORTS:

□ Report exactly identical recording titles

jSongMiner: Metadata miner

- Software for automatically acquiring formatted metadata about songs, artists and albums
- Designed for use with the Greenstone digital library software
  - May also be used for other purposes, such as cultural feature extraction
- Identifies music files
  - Uses Echo Nest fingerprinting functionality and embedded metadata
- Mines a wide range of metadata tags from the Internet and collates them in a standardized way
  - Data extracted from The Echo Nest, Last.FM, MusicBrainz, etc.
  - Over 100 different fields are extracted
  - Data may be formatted into unqualified and/or qualified Dublin Core fields if desired
- Saves the results in ACE XML or text
  - Can also be integrated automatically into a Greenstone collection

ACE XML: MIR research file formats

- Standardized file formats that can represent:
  - Feature values extracted from instances
  - Abstract feature descriptions and parameterizations
  - Instance labels and annotations
  - Class ontologies
- Designed to be flexible and extensible
  - Able to express types of information that are particularly pertinent to music
- Allow jMIR components to communicate with each other
  - Can also be adopted for independent use by other software
- ACE XML 2.0 provides even more expressivity
  - e.g. potential for integration into RDF ontologies

Previous collaborations (1/2)

- Vigliensoni, McKay and Fujinaga (2010)
  - Addition of Last.FM functionality to jWebMiner
  - Empirical comparison of different kinds of cultural data
- Angeles, McKay and Fujinaga (2010)
  - Addition of MusicBrainz functionality to jMusicMetaManager
  - Empirical comparison of curated, noisy and automatically cleaned metadata
- McKay, Burgoyne, Hockman, Smith, Vigliensoni and Fujinaga (2010)
  - Development of jLyrics and lyricFetcher
  - Empirical comparison of different feature types
- Thompson, McKay, Burgoyne and Fujinaga (2009)
  - Improvements to ACE

Previous collaborations (2/2)

- McKay, Burgoyne, Thompson and Fujinaga (2009)
  - Improvements to ACE XML
- McEnnis, McKay and Fujinaga (2006)
  - Improvements to jAudio
- Fiebrink, McKay and Fujinaga (2005)
  - An empirical investigation of dimensionality reduction using ACE (and other technologies)
- Sinyor, McKay, Fiebrink, McEnnis and Fujinaga (2005)
  - Beatboxing classification using ACE and jAudio

Paper ideas: Improve components

- jSongMiner
  - Add more data sources
- jAudio
  - Develop more features, especially psychologically meaningful features
  - Improve interface
- jSymbolic
  - Develop more features or implement the remaining feature catalogue
- jWebminer
  - Take advantage of additional web services (e.g. Amazon) to add more features
- jLyrics
  - Develop features especially relevant to music
- ACE
  - Add more machine learning algorithms
  - Especially unsupervised learning
- jMusicMetaManager
  - Add correction functionality

Paper ideas: Develop new components

- OMEN
  - Reimplement and get working
- jImage
  - Extract features from album art, press photos, etc.
- jVideo
  - Extract features from music videos, concert videos, etc.
- jMusicVisualiser
  - Offer visual ways of exploring relationships between musical instances, features and classes
- jStructure:
  - Automatically segment audio streams in time, both in terms of partitioning separate pieces of music within a single stream and in terms of structural divisions within individual pieces
- jClassOntology
  - Use data mining to harvest class ontologies

Paper ideas: Apply jMIR

- There are many problems to which the jMIR components could be applied
  - Either directly or with specialized improvements (e.g. features developed especially for chord recognition)
- Consider the many MIREX application areas
  - www.music-ir.org/mirex/wiki/MIREX_HOME

- More ideas:
  - See the Future Research sections of my papers and dissertation

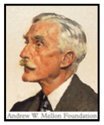
More information

- Overview, documentation and publications
  - jmir.sourceforge.net
- Code and (in most cases) manuals
  - sourceforge.net/projects/jmir/
  - sourceforge.net/projects/jaudio/
- My dissertation on jMIR
  - www.music.mcgill.ca/~cmckay/papers/musictech/mckay10dissertation.pdf
  - Other jMIR-related publications can also be found on my web page: www.music.mcgill.ca/~cmckay/
- E-mail
  - cory.mckay@mail.mcgill.ca

Acknowledgements

- Collaborators at McGill and Waikato Universities
- Funding, past and present:
  - CIRMMT
  - Centre for Open Software Innovation
  - Andrew W. Mellon Foundation
  - Social Sciences and Humanities Research Council of Canada
  - Canadian Foundation for Innovation
- Contact information:
  - jmir.sourceforge.net
  - cory.mckay@mail.mcgill.ca

Social Sciences and Humanities Research Council of Canada
Conseil de recherches en sciences humaines du Canada
Canada
nema
McGill
Schulich School of Music
École de musique Schulich
CIRMMT Centre for Interdisciplinary Research in Music Media and Technology
DDMAL
COSI Centre for Open Software Innovation
Greenstone



More details?

- jAudio: Audio feature extraction
- jSymbolic: Symbolic feature extraction
- jWebMiner: Cultural feature extraction
- lyricFetcher and jLyric: Lyric harvesting and feature extraction
- ACE: Meta-learning classification engine
- ACE XML: File formats
  - Features, feature metadata, instance metadata, ontologies
- Codaich, Bodhidharma MIDI and SLAC: datasets
- jMusicMetaManager and jSongMiner: Metadata management and harvesting

- General questions?