ACE: A General-Purpose Classification Optimization Framework

Cory McKay, Daniel McEnnis,
Rebecca Fiebrink and Ichiro Fujinaga
Music Technology Area, McGill University

Introduction

- Automatic music classification has many academic and commercial applications
- Many classification methodologies are available
  - Each has different strengths and weaknesses
- ACE experiments with a variety of approaches
  - Finds those appropriate for each classification problem
- ACE can then act as a basic classifier

Advantages of ACE

- Designed to meet particular needs of music researchers
  - Can assign multiple classes to individual recordings
  - Deals naturally with windowed and segmented music
  - Allows structured hierarchical taxonomies
  - Allows multi-dimensional features
- Uses and evaluates many algorithms
  - Includes classifier ensembles
  - Evaluates speed as well as accuracy
- Can be easily ported to arbitrary feature extractors
  - Also bundled with audio and MIDI feature extractors
- Easy-to-use interface
  - Includes on-line help
- Open source and easily extensible

Overview of ACE

Potential users

- General users
    - Musicologists, theorists, librarians, psychologists, etc.
    - No knowledge of underlying machine learning needed
    - Simple graphical interface
    - See *Applications of automated classification* section below
- Users knowledgeable in pattern recognition
    - Can evaluate new classifiers and features
        - Provides a baseline for new algorithms
    - Good development environment
        - Implemented in Java
        - Integrated with Weka

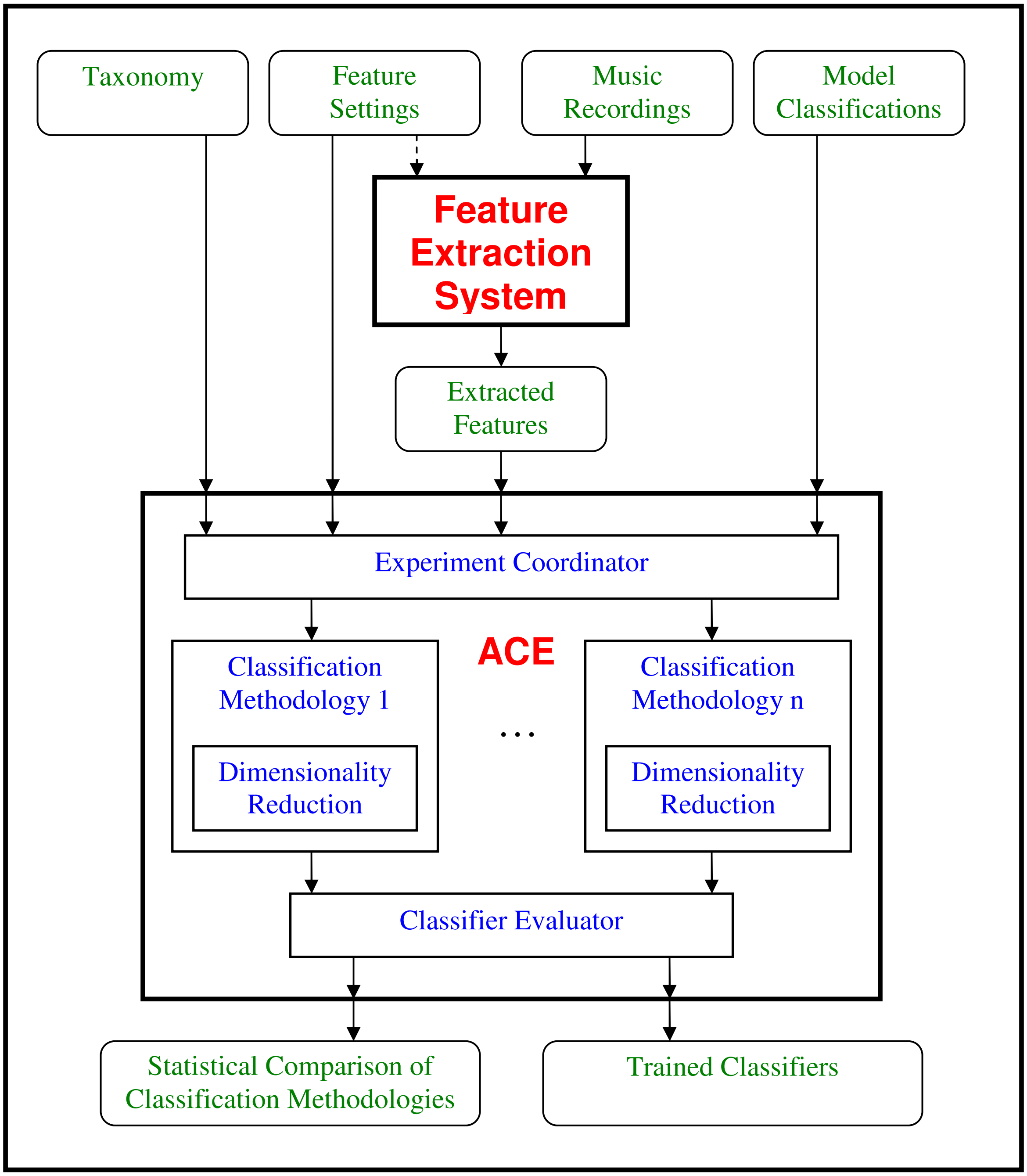
Input to ACE

- Feature vectors (Weka ARFF or ACE XML)
- Model classifications for training and testing (optional)
- Taxonomy (optional)

Output from ACE

- Comparison of different classifiers
- Feature effectiveness measurements
- Trained classifiers

Graphical representation of ACE



Importance of music classification

Overview of music classification

- Many ways to classify music
    - Genre, composer, performer, mood, emotional content, geographical or temporal origin, listening scenarios, etc.
- Classification often difficult for both humans and computers
    - Rarely have precise, clear and consistent guidelines delineating the characteristics of categories

Advantages of computer classification

- Fast and consistent
- Can analyze music in novel and non-intuitive ways that might not occur to humans
- Avoids human preconceptions that could contaminate results

Applications of automated classification

- Sorting of large music databases
    - Human classification is slow and expensive
    - Also applicable to personal collections
- Music recommendation systems
- Interactive accompaniment systems
- Automated transcription
- Detection of pirated recordings
- Composer identification of anonymous pieces
- Studies of relative feature and taxonomy appropriateness
- Research into how humans perceive musical similarity and form musical groupings

Introduction to automatic music classification

Feature extraction

- Features are characteristics that can be used to distinguish between different types of music
- Feature extraction is the process of extracting features from music
- Features serve as the input to classifiers
    - Good features are essential to classification

Types of recordings

- Audio recordings are reproductions of actual sound
    - MP3, .wav, .aiff
- Symbolic recordings consist of high-level musical instructions rather than actual audio samples
    - MIDI, Humdrum

Classifiers

- Classifiers are software modules that attempt to distinguish between different categories or classes
- Classifiers may be implemented using:
    - Expert systems: use pre-programmed heuristics
    - Machine learning: uses pattern recognition algorithms to “learn” to identify classes
- Meta-learning: system finds good classifiers to use for a particular problem
    - This is what ACE does

Existing classification systems

Music-oriented systems

- Usually implemented with particular tasks in mind
  - Not extensible to general music classification tasks
  - Utilize limited techniques
  - Difficult to use
- Need for standardized systems
  - Avoids reimplementation
  - Well tested and reliable code
  - Better interfaces and more usable software
  - Facilitates methodology comparisons
- Important steps in this direction:
  - M2K (Downie 2004)
  - Marsyas (Tzanetakis & Cook 1999)
- Often lack powerful techniques found in general systems

General systems

- Commonly used frameworks:
  - Weka (Witten & Frank 2005)
  - PRTools (van der Heijden et al. 2004)
- A few use meta-learning:
  - Metal (www.metal-kdd.org)
  - AST (Lindner and Studer 1999)
- Often require proprietary software, are not open source or have licences limiting commercial application
- Not designed to meet the particular needs of music

Specialized needs of music classification

Qualities music classification systems should have

- Be able to assign multiple classes to single recordings
  - A song could belong to multiple genres, for example
- Allow overall classification of recordings as well as of individual sub-sections
  - Audio often windowed
  - Essential to segmentation problems
- Maintain logical grouping of multi-dimensional features
  - Musical features often consist of vectors
    - e.g. MFCC's
  - Provides classification opportunities
- Maintain recording meta-data
  - Title, performer, composer, date, etc.
- Take advantage of hierarchically structured taxonomies
  - Provides classification opportunities

ACE meets these needs

- Integrated into machine learning engine
- Integrated into graphical interface
- New standardized XML file formats proposed
  - Existing standards (e.g. Weka's ARFF) insufficient
  - Flexible and human readable
  - Designed to allow reuse of files for different projects

ACE's classification methodology

Implementation

- Uses meta-learning to find approaches well suited to particular problems
- Built on the Weka library
  - Easy to add new classifiers

Classifiers used

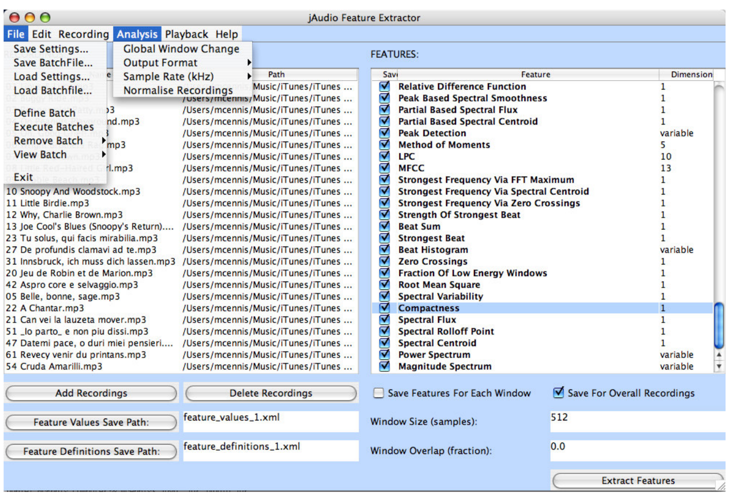
- Base classifiers include:
  - Induction trees
  - Naive Bayes
  - Nearest neighbour
  - Neural networks
  - Support vector machines
- Classifier parameters are varied
  - Neural network architecture
  - Value of k in k-NN classifier
  - etc.
- Classifier ensembles utilized
  - Multiple experts may perform better than one
  - Bagging, boosting, etc.
  - Known to be powerful tools
    - Rarely applied to music to date
- Dimensionality reduction also used
  - Principal component analysis
  - Feature selection using genetic algorithms
  - Exhaustive search

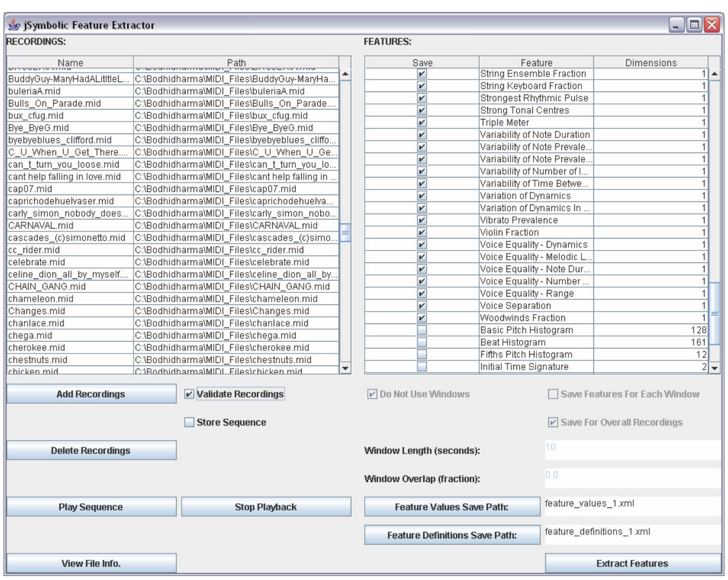
Feature extraction

Overview

- ACE usable with arbitrary feature extractors
    - Reads Weka ARFF and ACE XML files
- Bundled with two powerful and extensible feature extractors
    - jAudio: for use with audio recordings
        - McEnnis et al. 2005
    - jSymbolic: for use with MIDI recordings
        - Based on Bodhidharma (McKay 2004)

Interfaces of jAudio and jSymbolic





Interfaces

Command line interface

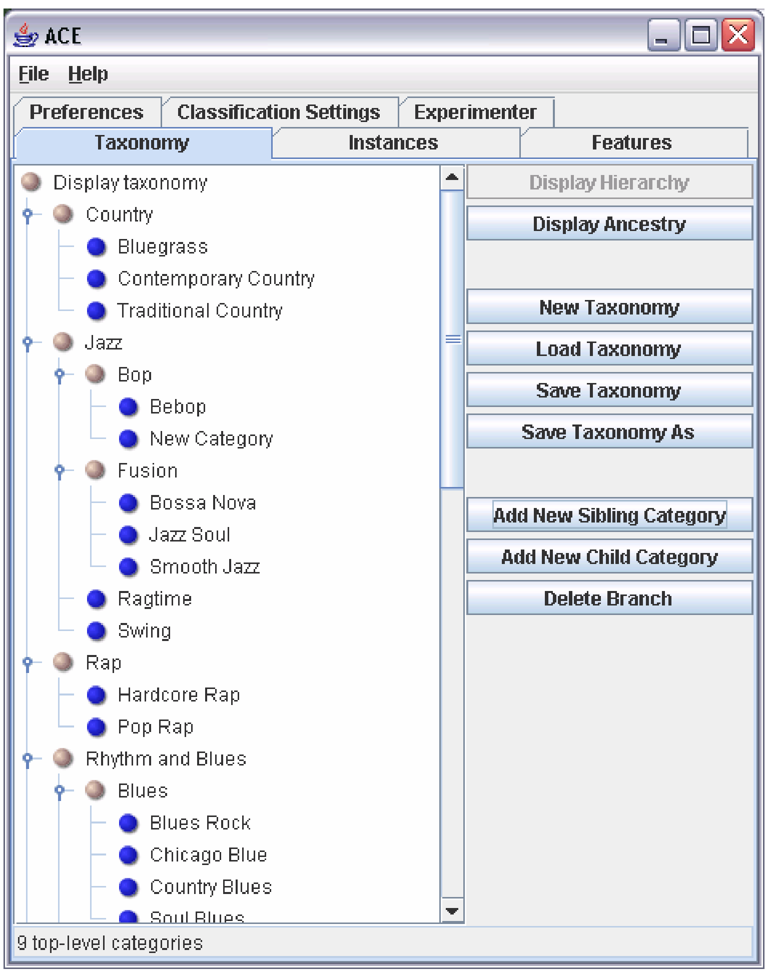
- Batch processing

Java API

- Open source and well documented

Graphical interface

- Includes an on-line manual
- Can build taxonomies, label and manage recordings, manage features, control classifier settings, carry out comparisons of classification methodologies and train and use classifiers



Experimental evaluation

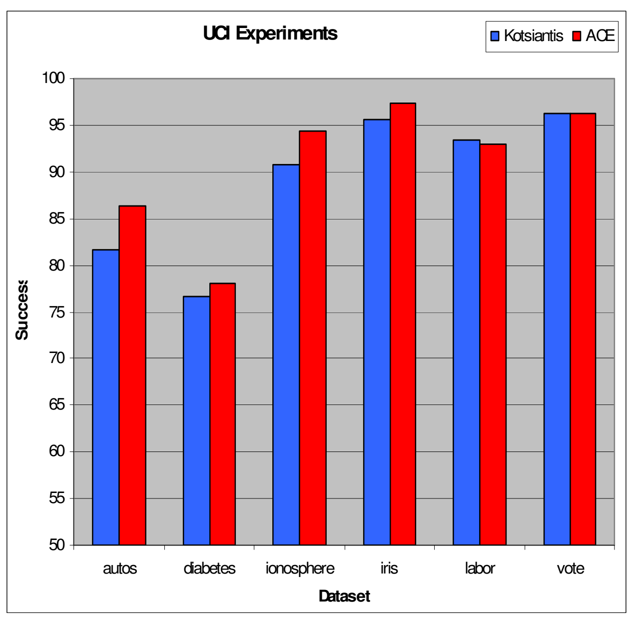
Musical experiments

- Repeated Tindale's drum recognition experiment (2004)
  - ACE achieved 96.3% success compared to Tindale's best rate of 94.9% (error rate reduction of 27.5%)
- Achieved 95.6% success with a 5-class beatbox recognition experiment (Sinyor et al. 2005)

General experiments

- Applied ACE to six UCI datasets
- Compared results with a successful recently published algorithm (Kotsiantis and Pintelas 2004)
- ACE performed better in most cases
  - ACE limited to only one minute per learning scheme
- Different classifiers performed better with different datasets
  - No classifier always better than others
    - Supports ACE's experimental approach
  - Effectiveness of AdaBoost (chosen 4 times out of 10) shows utility of classifier ensembles

Data Set	ACE's Selected Classifier	Kotsiantis Success Rate	ACE Success Rate
autos	AdaBoost	81.7%	86.3%
diabetes	Naïve Bayes	76.6%	78.0%
ionosphere	AdaBoost	90.7%	94.3%
iris	FF Neural Net	95.6%	97.3%
labor	k-NN	93.4%	93.0%
vote	Decision Tree	96.2%	96.3%



Project status

Timeline

- Currently in alpha release
- Full release: January 2006
    - Will include finalization of GUI

Long-term goals

- Use distributed computing to decrease processing times
    - M2K/D2K or Grid Weka
    - Will use idle time to improve inactive projects
- Have ACE keep track of past experiments
    - Use data to guide strategies in future projects
- Automatic formation of taxonomies using clustering
- Incorporate blackboard systems
    - Will include the possibility of using expert systems
- Post-processing modules

Contact information

- Web site: coltrane.music.mcgill.ca/ACE
- E-mail: cory.mckay@mail.mcgill.ca

McGill

Social Sciences and Humanities Research Council of Canada

Centre for Interdisciplinary Research in Music Media and Technology

Fonds de recherche sur la société et la culture Québec