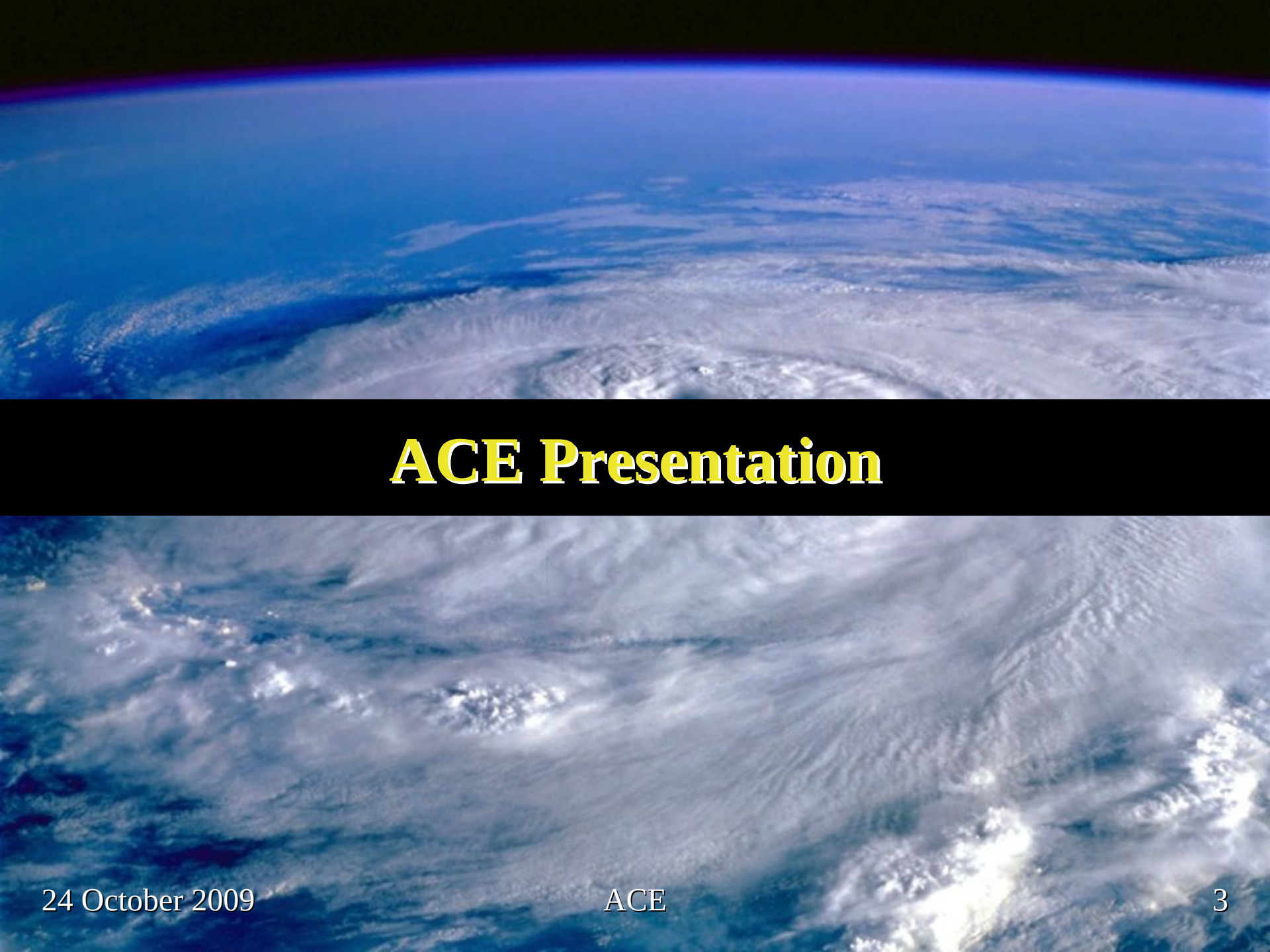

ACE

Contents

- ACE Presentation
- Comparison with existing frameworks
- Technical aspects
- ACE 2.0 and future work



ACE Presentation

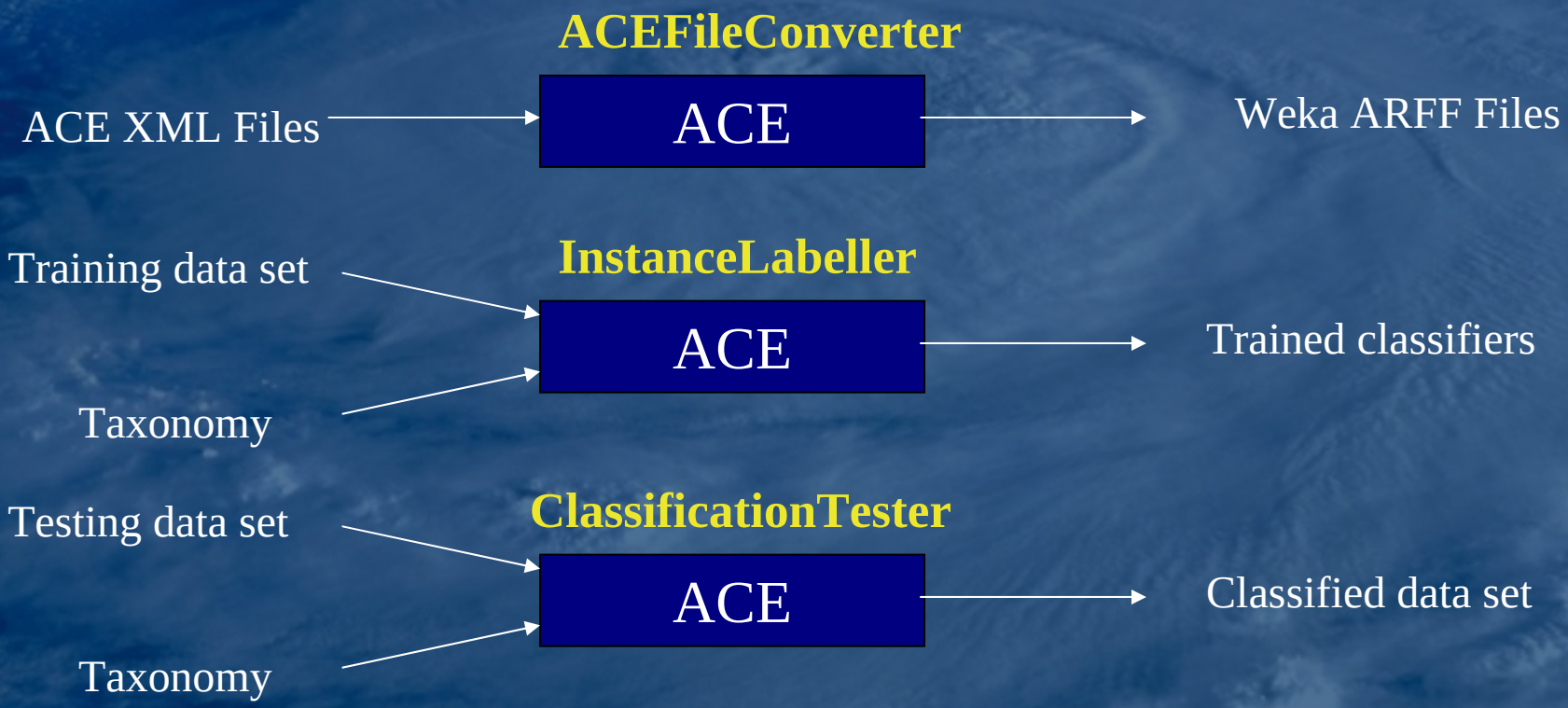
ACE Presentation

- Framework for using and optimizing classifiers
- Experiments a variety of classifiers, classifier parameters, classifier ensembles and dimensionality reduction
- Designed to facilitate classification

➔ **Meta-learning component**

ACE Configurations

- ACE 1.1 works with command line interface
- ACE 2.0 works with command line or GUI interface



Example

(Devaney 2008)

DIMENSIONALITY REDUCTION: Genetic search using naive Bayesian classifier
TIME TAKEN: 0.40363333333333334 minutes
SELECTED FEATURES (12 of 24):

- Spectral_Centroid_Overall_Standard_Deviation
- Spectral_Rolloff_Point_Overall_Standard_Deviation
- Spectral_Flux_Overall_Standard_Deviation
- Compactness_Overall_Average
- Root_Mean_Square_Derivative_Overall_Average
- Zero_Crossings_Overall_Standard_Deviation
- Zero_Crossings_Derivative_Overall_Average
- Zero_Crossings_Derivative_Overall_Standard_Deviation
- Strongest_Frequency_Via_Zero_Crossings_Overall_Average
- Strongest_Frequency_Via_Spectral_Centroid_Overall_Average
- Strongest_Frequency_Via_Spectral_Centroid_Overall_Standard_Deviation
- Strongest_Frequency_Via_FFT_Maximum_Overall_Average

Example

(Devaney 2008)

BEST CLASSIFIER: AdaBoost with C4.5 Decision Trees
BEST AVERAGE ERRORRATE: 6.2080536%
CORRESPONDING CROSS-VALIDATION TIME: 0.23671666666666666 minutes
FOLDS: 10

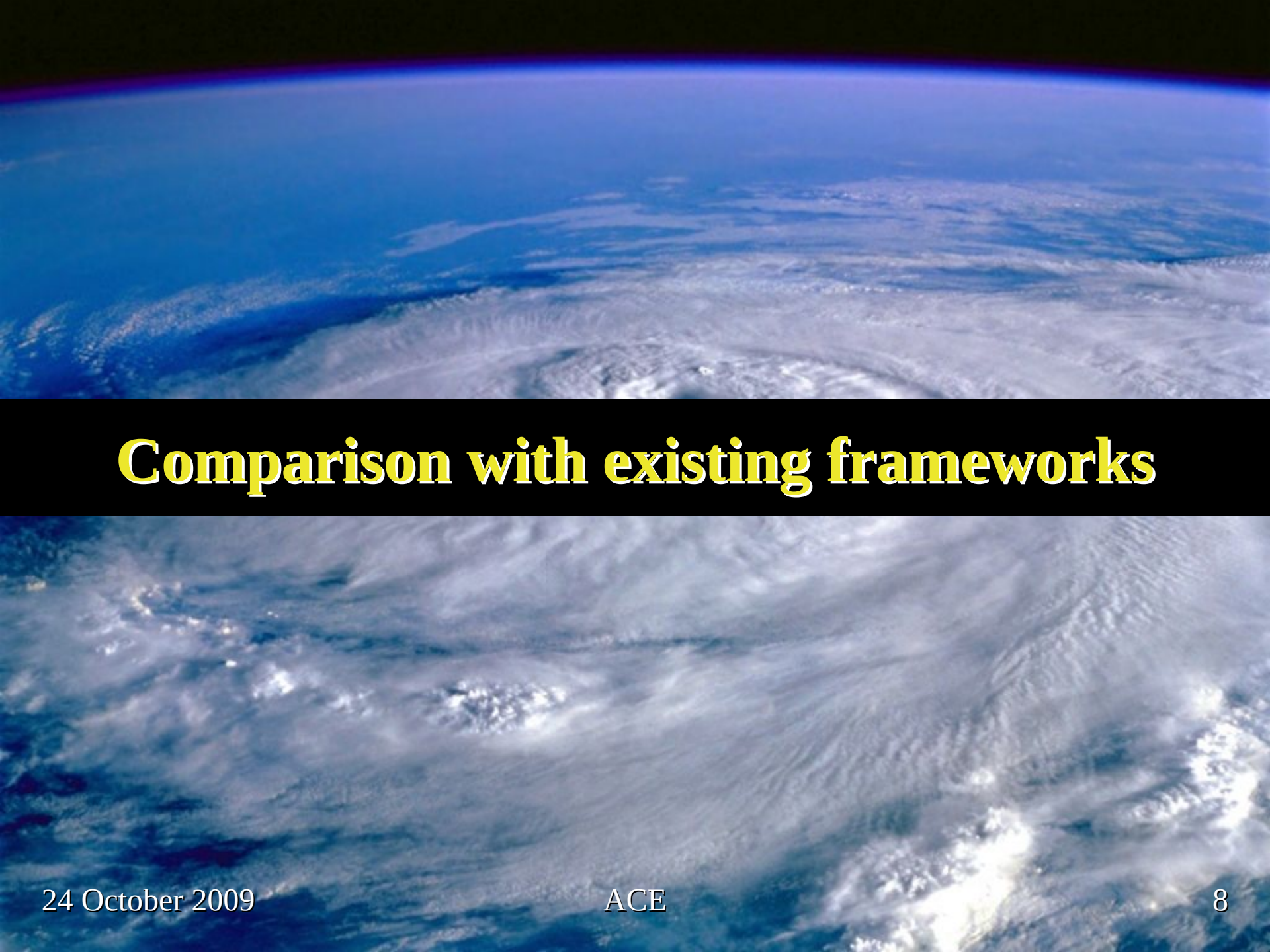
CROSS-VALIDATION STATISTICS:

Correctly Classified Instances	1118	93.7919 %
Incorrectly Classified Instances	74	6.2081 %
Kappa statistic	0.9204	
Mean absolute error	0.0257	
Root mean squared error	0.1553	
Relative absolute error	8.2325 %	
Root relative squared error	39.3154 %	
Total Number of Instances	1192	

Ignored Class Unknown Instances 49

CONFUSION MATRIX:

a	b	c	d	e	<-- classified as
308	0	2	0	1	| a = kick
0	263	26	0	1	| b = open
2	23	264	7	2	| c = closed
0	0	4	150	2	| d = k-snare
2	1	1	0	133	| e = p-snare



Comparison with existing frameworks

Existing general frameworks

- PRTTools (MATLAB Toolbox) – *Van der Heijden et al. 2004*
  - Reliant upon a proprietary software
  - Non redistributable
  - Little portability
- Weka – *Witten and Frank 2000*
  - Freely distributable
  - Open source
  - Relatively well documented
  - Good portability (implemented in Java)

➔ Basis of the ACE engine

Existing specific frameworks

- Marsyas - *Tzanetakis and Cook 1999*
  - C++ based system – little portability
  - No MIDI classification
  - Designed as feature extractor
- M2K - *Downie 2004*
  - Feature extraction and classification framework based on D2K
  - License issues
  - Not open source

Technical aspects

Feature file formats

(McKay et al. 2005)

- ACE is designed to be used with any feature extractor and data set correctly formatted

```
<!ELEMENT feature_vector_file (comments,
data_set+)>
<!ELEMENT comments (#PCDATA)>
<!ELEMENT data_set (data_set_id,
section*,
feature*)>
<!ELEMENT data_set_id (#PCDATA)>
<!ELEMENT section (feature+)>
<!ATTLIST section start CDATA ""
stop CDATA "">
<!ELEMENT feature (name, v+)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT v (#PCDATA)>
```

- Weka ARFF file format
  - 1 class = 1 instance
  - Grouping of features
  - Labeling or structuring of the instance
  - Structure of class labels

- ACE XML file format
  - Easily readable (verbose format) and standardized
  - 1 file = 1 task (features, feature definitions, classifications…) ➔ reusable
  - Metadata storage ➔ full independence between extractors and classifiers

Meta-learning - Classifier ensembles

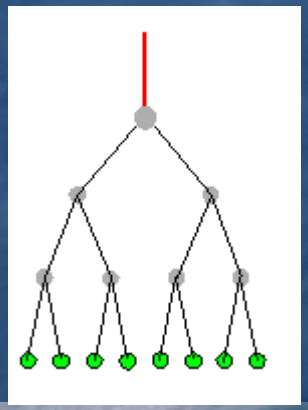
- Combine the answer of several classification models
- Advantages (*Dietterich 2000*)
  - Statistical performance
  - Improved results for models working with local optima
  - Can improve significantly results from a non adapted format classifier
- Classifier ensemble methods
  - Voting (merging the results)
  - Dynamic selection (selecting in each case the best classifier)
  - Stacking (weighted voting)
  - Bagging (sampling all available training instances)
  - Boosting (emphasize poorly classified trainings sets)

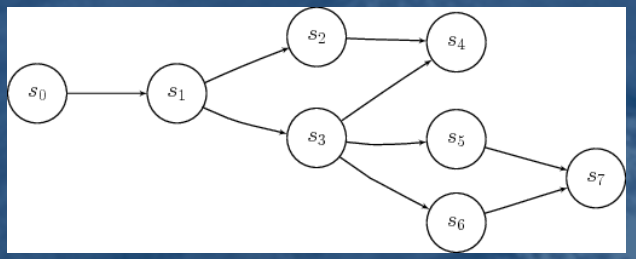
ACE Classifiers

- Feedforward neural networks
- Support vector machines
- Nearest neighbour classifiers
- Decision tree classifiers
- Bayesian classifiers

Feature dimension reduction

- ACE implements a variety of dimensionality reduction and feature weighting techniques
- Iterative evaluation of the techniques
- Techniques used:
  - Principal component analysis
  - Genetic algorithms
  - Tree searches
  - Forward-backward algorithms





Feature weighting with GA's

(Fiebrink et al. 2005)

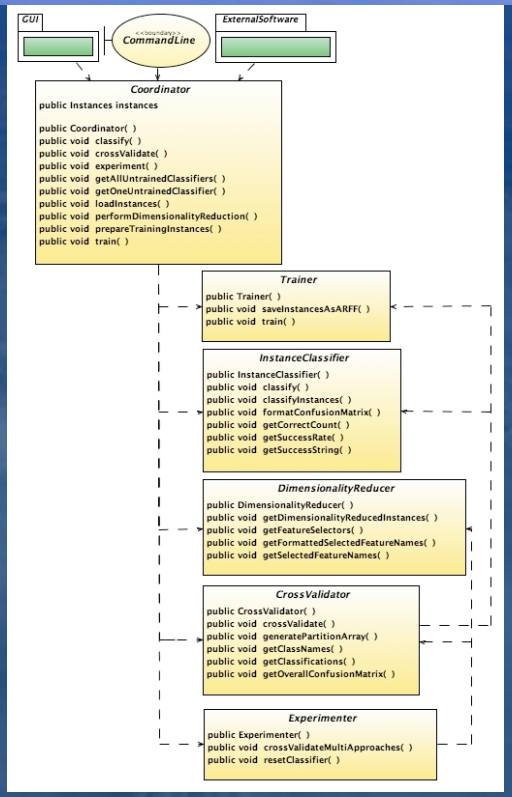
- Optimize a classification by assigned weights to the features according to their accuracy
- GA software: JGAP (Java package)
- Genes selection: leave-one-out cross-validation with k-NN classifier
- Parallel computation (M2K/D2K or Grid Weka)

ACE 2.0 and future work

ACE 2.0

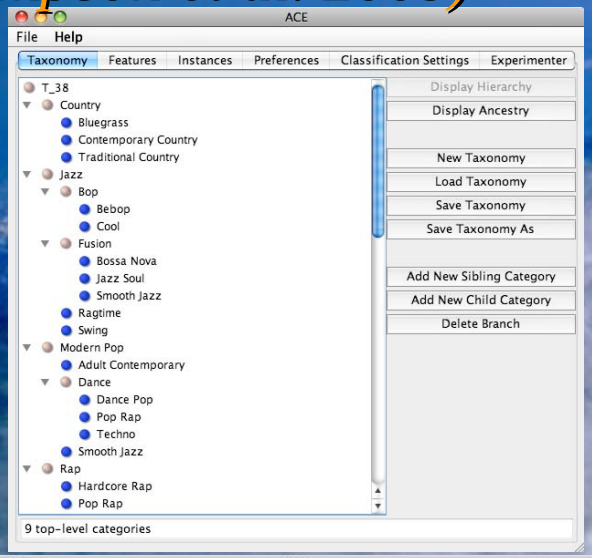
(Thompson et al. 2009)

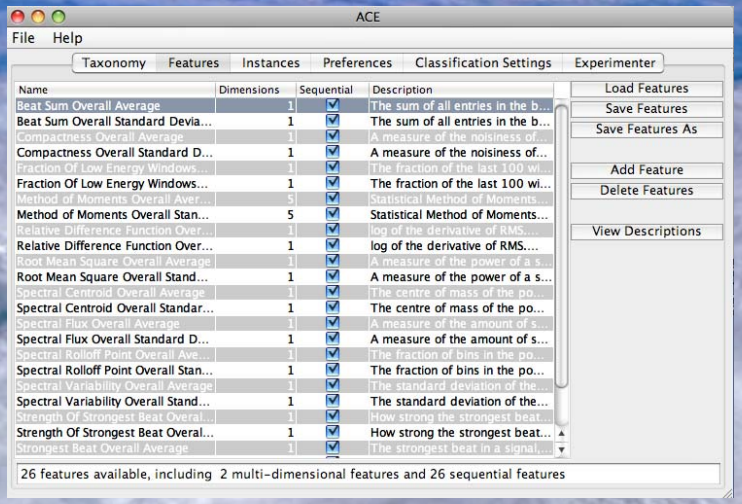
- Architectural restructuring
  - ➔facilitate integration with other softwares
- Redesigned cross-validation
  - ➔Implemented in ACE rather using Weka
  - ➔More transparent processing
- ACE XML 2.0 Zip and Project files
  - ➔Combine the different ACE XML files
- Improved command-line interface
- Graphical User Interface
  - ➔viewing and editing ACE XML files

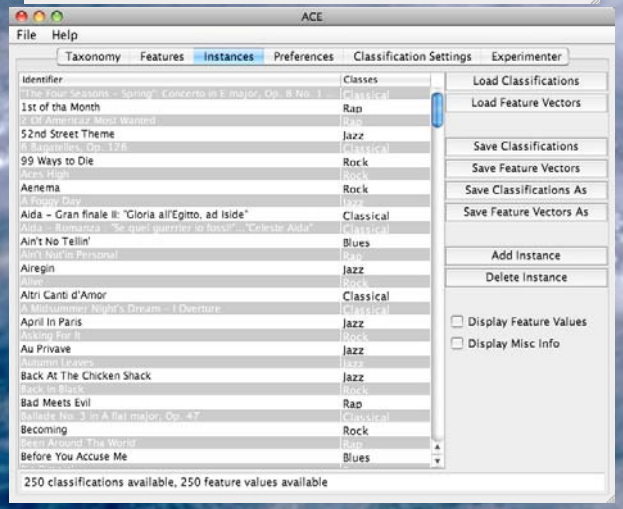


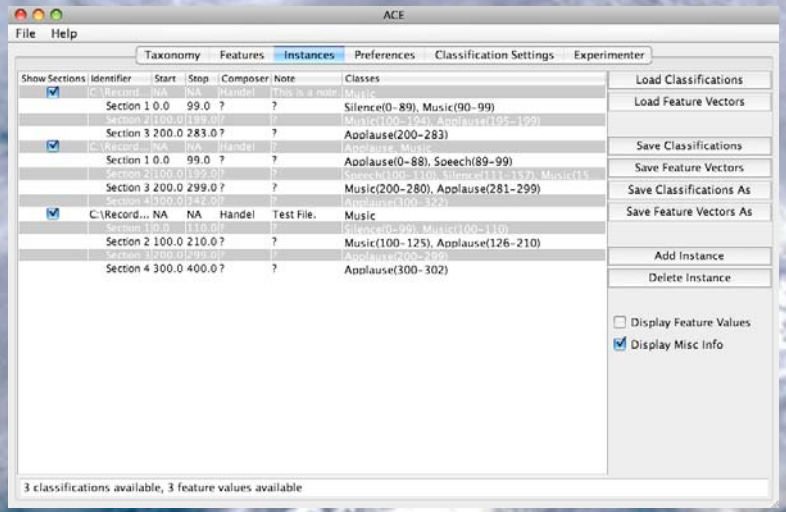
GUI ACE 2.0

(Thompson et al. 2009)









Future Work on ACE

(Thompson et al. 2009)

- Fully functional GUI
- Distributed work load
- Expanding learning algorithms
  - HMM
  - Recurrent neural networks
  - Weka's unsupervised learning

Conclusion

- User-friendly and very complete framework
- Allows the user to test his/her own classifiers, data reduction techniques…
- Encourage experimentation in classification, in particular with classifier ensembles

Table 1. (McKay et al. 2005) ACE's classification success rate on ten UCI datasets using ten-fold cross-validation com-pared to a published baseline (Kotsiantis and Pintelas 2004).

Data Set	ACE's Selected Classifier	Kotsiantis Success Rate	ACE Success Rate
anneal	AdaBoost	--	99,6%
audiology	AdaBoost	--	85,0%
autos	AdaBoost	81,7%	86,3%
balance scale	Naïve Bayes	--	91,4%
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%
zoo	Decision Tree	--	97,0%