NIPS 2008 Workshop
Machine Learning Open Source Software

“BenchMarking Via WEKA”

Peter Reutemann
fracpete at waikato dot ac dot nz
What is WEKA?

- **Waikato Environment for Knowledge Analysis**
- Machine learning work-bench written in Java

“WEKA Experimenter”

Datasets

Schemes

Key for Scheme setups
Motivation

• WEKA's benefits
  • plug-in architecture for schemes
  • framework for statistical evaluation/comparison
  • experiments are relatively easy to reproduce

• WEKA's limitations
  • written in Java, limited support for Python (via Jython)
  • no benchmarks of standard schemes publicly available

How to integrate other languages/frameworks, but keep benefits of centralized statistical framework and easily reproduce experiments???
The BMVW Framework

• Client-server architecture
• Communication via XML protocol
  ⇒ programming language agnostic
• Clients:
  • obtain dataset from server
  • build scheme
  • send predictions back
• Server evaluates predictions and stores results

- Datasets
- Experiment setups
- Evaluation
- Results

Server [XML] Client(s)

- WEKA (Java)
- MLPY (Python)
- etc.
Reproducibility of Experiments

• Datasets need to be uploaded to server
  ⇒ Publicly available

• Single experiment is uniquely identified by:
  • Type of evaluation (cross-validation, percentage split, etc.)
  • Scheme setup (name, parameters, version)
  • Dataset (name, version)

• Controlled, centralized evaluation:
  • Server generates training and test data for client
  • Server calculates statistics based on predictions from client
Experiments Screenshot

Datasets × Schemes matrix

Datasets

Key for Scheme setups
Conclusion

• Step towards straight-forward experiment reproducibility

• Evaluation framework for different tools across programming languages (WEKA, MLPY, ...)

• Generation of ExpML output easy

• Submission of results to Experiment Database easy
http://www.scms.waikato.ac.nz/~fracpete/projects/bmvw/

...thank you!