Data Mining I

Classification Workflow with Rapidminer
Outline

1. Data Import
2. Preprocessing
3. Classification
4. Evaluation
Data Import

- Import your data into Rapidminer Repository
  - Everything in one place
  - Valuable meta-data for further processing

- Use the import wizard, if available
Preprocessing

• Look at your data
  • What is the target attribute?
    • Is the target attribute already a label?
  • What is the distribution of labeled examples by class?
    • Is my classifier capable of handling imbalanced data?
  • What other attributes are available?
    • Is my classifier able to handle these types of attribute?
  • What are the ranges of the attributes?
    • Is my classifier good in handling various ranges?
  • What attributes correlate?
    • Is my classifier able to handle strongly correlating attributes?

–See Exercise 1 for more information.
Set Roles & Normalization

- Set roles for attributes
- Normalize attribute values
Discretize

- Numerical attributes can be divided into bins using discretization
- By Size (equally sized data ranges per bin)

- By Frequency (equally sized number of examples per bin)
Balancing

- Sampling (with balancing)

- Multiplication of data
  - Filter under-represented class examples
  - Append them to original example set
Classification

• Input: data set with labels
• Output: classification model

• Known Classifiers:
  • K-NN
  • Naive Bayes
  • Decision Tree (Hunts & ID3)
  • Rule Induction & Tree to Rules
  • Support Vector Machine (libSVM)
  • Neural Networks
Evaluation

- Evaluate on dedicated test data set

- Evaluate on one data set using
  - Split validation
  - X-Validation
Split-/Cross-Validation

- Split-validation is a *holdout method*, which reserves a certain amount for testing and uses the remainder for training.
  - First step: split data at a ratio in test and training set
  - Second step: learn a model on the training set and evaluate the model on the test set

- *Cross-validation* avoids overlapping test sets
  - First step: data is split into $k$ subsets of equal size
  - Second step: each subset in turn is used for testing and the remainder for training

Important: Never ever use the same example for training & testing!
Performance

- Standard Measures
  - Accuracy
  - Precision
  - Recall

- Task Specific
  - Misclassification Costs
Questions?