# Introduction to Machine Learning CS436/580L
Weiying Dai, Fall 2016
Homework Assignment 3
Out: Oct 23, 2016
Due: at 11:59pm on Nov 4, 2016 (Friday)

## 1. Overview of Assignment

We have studied theories about model overfitting and model selection, and discussed practical procedures of model selection. In this assignment, we will gain hands-on experience of model selection on a real-world classification problem using a popular machine learning software package Weka.

The next two sections provide some background for the assignment. The required tasks are described in Section 4. In addition, an optional task for bonus credit is given in Section 5.

## 2. Background on the Classification Problem

In this assignment, we are going to predict whether a mushroom is poisonous or not (edible). We use a noisy mushroom data set for this problem. Using this data set, we will train decision trees and Naive Bayes classifiers to classify each mushroom as poisonous or not, using 22 discrete features such as cap shape, cap color and gill size.

The original data set is available at the UCI machine learning repository.
http://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/

Detailed description of the data and its past usage is at
http://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/agaricus-lepiota.names

The original data set has been preprocessed by removing instances of missing values, adding noise to the feature values (changing some of the feature values at random), and converting to the arff format required by Weka. The preprocessed data has been randomly divided into two data sets: mushroom-train.arff (the training set) and mushroom-test.arff (the independent test set), which will be used for this assignment.

## 3. Background on Weka Software

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

- Download and install Weka3.6

  On weka homepage (http://www.cs.waikato.ac.nz/ml/weka/), click on "download", and choose to download and install any version of Weka 3.6 based on your operating system.

- Run Weka3.6 from GUI

  On the GUI interface, run "Explorer". A GUI called "Weka Explorer" should jump out.

  On Weka Explorer interface, to read in a data set, click "Open file" under "Preprocess".

  After a data set is successfully loaded, try out some classification algorithms under "Classify".

- Useful references and resources

  On Weka homepage, you can click on "documentation" to get help and more information about using Weka. You don't really need to read the entire documentation to get around of the functions of Weka.

  In "General documentation" section:
  - Weka API: Weka API can help you know how the classes are organized and how to call the methods from each class
  - Weka mailing list with searchable archive: you can post your questions on the mailing list. People who have experience coding with Weka will answer your questions.

  In "Tutorial" section:
  - A set of ppt slides which provides comprehensive information about how to use Weka from GUI interface

  In "Technical" section:
  - Document that explains ARFF data format required by Weka

- Classifiers information

The decision tree and Naive Bayes classifiers we learned in class are available in Weka. In particular, a version of the C4.5 decision tree algorithm is called J48 and is under the directory of trees. The Naive Bayes classifer (NBC) introduced in class is called NaiveBayesSimple and is under the directory of Bayes.

## 4. Required Tasks

### Task 1: Train decision tree without pruning and observe model overfitting (20 points)

Use the provided training data to build a fully-grown J48 tree (set the "unpruned" option to True). Then apply the tree on the provided test data. Report the following:

1) The size of the tree (in terms of total number of nodes)
2) The training error $error_D$
3) The test error $error_S$
4) The 95% confidence interval of the true error on future data $error_{\mathcal{D}}$

### Task 2: Decision pruning and model selection by cross validation (55 points)

Build pruned J48 trees (set the "unpruned" option to False) by setting the values of the pruning control parameters, and select the best setting based on cross validation using only the training set.

There is a number of pruning control parameters in the J48 implementation. In this task, you only need to experiment with the following two and leave the rest at the default settings.

- Minimum number of instances in a leaf node (MinNumObj). This is a pre-pruning criterion that can be combined with the reduced error pruning option below.
- With the reduced error pruning option set to True, the number of folds (numFolds) used to further divide the training data into training and validation folds

Set the two parameters at different values (it's your choice to specify a range of values for each parameter). For each setting, report the average error of the 10 pruned trees resulted in 10-fold cross validation. Note that this loop of 10-fold cross validation is different from the inner loop of n-fold cross validation (n is the parameter to set) used by reduced error pruning. For clarity of presentation, you should report the results in tables or figures. (30 points)

Decide the best parameter setting (corresponding to the best model complexity) based on the reported results, and use that setting to retrain a mode using the entire training data (mushroom-training), and test the model on the provided test data (mushroom-test). Report the following: (25 points)

1) The parameter setting used to retrain a J48 tree based on the entire training data
2) The size of the tree (in terms of total number of nodes)
3) The training error $error_D$
4) The test error $error_S$
5) The 95% confidence interval of the true error on future data $error_{\mathcal{D}}$

**Task 3: Compare learning algorithms by cross validation (25 points)**
In task 2, you have done model selection among different trees. In this task, you are going to perform model selection across different learning algorithms, particularly, to choose between decision tree and NBC.

Report the average error of NBC models resulted in 10-fold cross validation. (5 points)

Compare this average error with the average error produced by the best setting of J48 in 10-fold cross validation from Task 2. Decide which algorithm is better for mushroom classification on future data. Justify your choice without using the test data. (Hint: is the difference in the two average errors statistically significant?) (10 points)

Now use the test data to validate your choice.
Report the test error $error_S$ of the finally selected model on the test data (5 points)
Report the test error $error_S$ of the other model on the test data (5 points)
Does your choice actually predict better on the independent test data (Does model selection work for this classification problem)?

**5. Optional Task (50 bonus points)**

Implement an experimenter (batch processing) program which automatically performs the process of model selection for Task 2 and Task 3.

The program should be able to take the following as input:

- Training and test sets
- Learning algorithms to be considered (in this task, J48 and NaiveBayesSimple)
- Different parameter values for J48 pruning

The program should produce at least the following in the output:

- The average error of 10-fold cross validation for each parameter setting of J48
- The average error of 10-fold cross validation for NaiveBayesSimple
- The best model produced on the entire training set and its associated parameters
- The classification of each test instance by the best model
- The test error $error_S$ of the best model on the test set

To minimize effort of programming, your program should utilize the classification algorithms and other functions such as data input and cross validation provided by Weka.

***Note: The due date of HW3 only applies to the required tasks. The submission for the optional task is due until 11:59pm of Nov 17.***