COC131 Data Mining - Classification I

Martin D. Sykora m.d.sykora@lboro.ac.uk
Tutorial 03, Friday 7th March 2008

A (very) brief overview of classification

The aim of classification is to be able to predict the *discreet value* of a class based on a set of real-valued or discrete attributes. The aim of regression is to be able to predict the *real value* of a dependent variable based on a set of real-valued or discreet independent variables. So a trained classifier is simply a function of the form:

$$f(\mathbf{x_a}, \mathbf{w_a^*}) \Rightarrow \mathbb{Z}$$

in the case of a classification problem, or

$$f(\mathbf{x_v}, \mathbf{w_v^*}) \Rightarrow \mathbb{R}$$

in the case of a regression problem. Where $\mathbf{x_a}$ is a vector of attributes, $\mathbf{x_v}$ is a vector of independent variables, $\mathbf{w_a^*}$ is a vector of optimum classification model parameters, $\mathbf{w_v^*}$ is a vector of optimum regression model parameters, $\mathbb{Z}$ is the set of integers and $\mathbb{R}$ is the set of real numbers. The underlying function will depend on the type of classifier used. In order to determine the optimal model parameters the model must be trained. Typically this involves iteratively adjusting the model parameters to minimize the model error on a training set. Once the model error drops below a certain threshold training is stopped and $\mathbf{w}^*$ is the current set of model parameters.
When training a classifier on a particular data-set it is important to avoid over-fitting. This is the process of creating a classifier which is overly specialised on the training data i.e. it performs well on the data it is trained with, but very poorly on a new unseen data-set. This is done by using a test set to test the performance of the model. As the model is trained both the training error and testing error will decline, but eventually the test error will

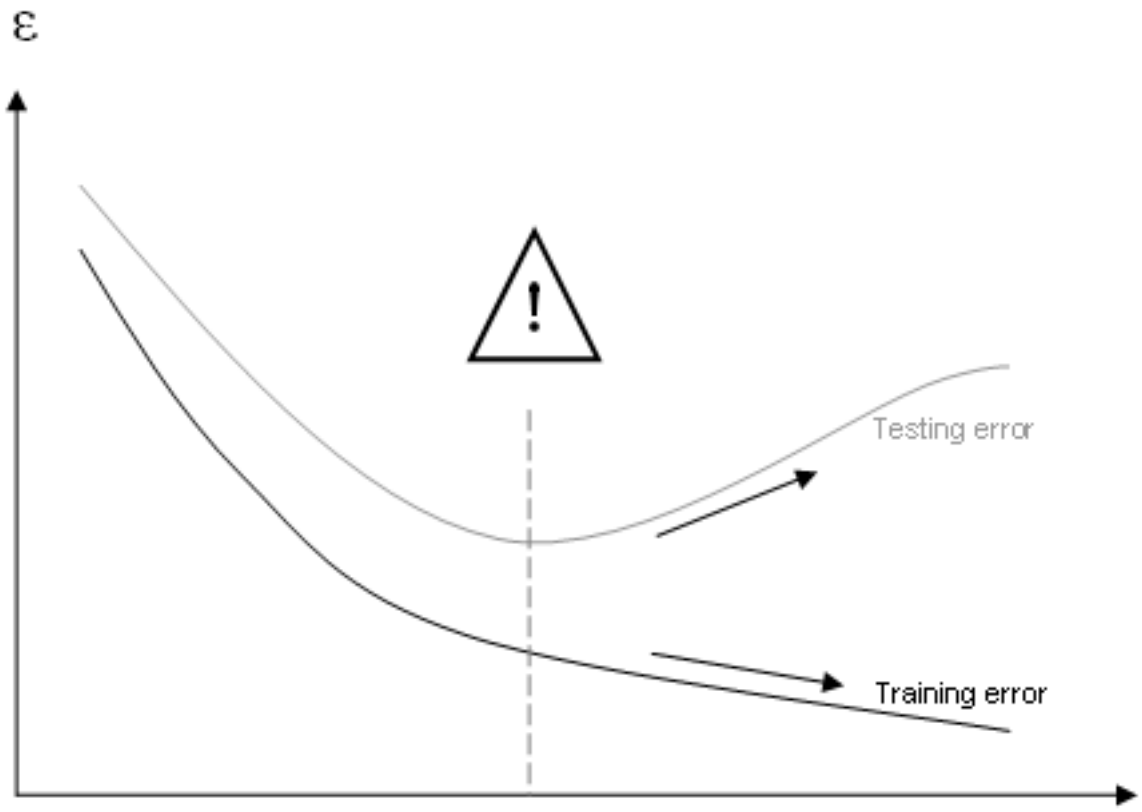

Figure 1: Over-fitting

start to increase as the model becomes over-fitted, at this point the training process should be stopped (Figure 1).

The distinct stages of designing a classification model are outlined below:

- Collect your raw data
- Clean your data (e.g. outlier removal, missing data removal etc.)
- Preprocess the data (e.g. normalization, standardization, etc.)
- Determine the type of problem (i.e. classification or regression)
- Pick an appropriate classifier (e.g. multilayer perceptron, decision tree, linear regression, etc.)
- Choose some default parameters for the classifier, the choice of classifier and parameters constitute your model

- Pick a training/testing strategy (e.g. percentage split, cross-validation etc.)
- Train the classifier using your training/testing strategy
- Analyse the performance of your model
- If your results are unsatisfactory consider altering your model (i.e. changing the classifier, its parameters, and/or your training/testing strategy) and re- training/testing
- If your results are satisfactory validate your model on an unseen set of cleaned and preprocessed data.

Exercises

1. Fire up the Weka (Waikato Environment for Knowledge Analysis) software, launch the explorer window and select the "Preprocess" tab. Open the iris data-set ("iris.arff", this should be in the ./data/ directory of the Weka install).
2. Select the "Classify" tab. Under the "Test options" section you have four different testing options. How do each (we cannot use "supplied test set" option as we have no applicable file) of these options affect the training/testing? Which testing mode do you think will perform best? (the file "ExplorerGuide.pdf", page 10, in the ./ directory of the Weka install may help).
3. Under "Classifier" select "MultilayerPerceptron". What type of classifier is this? How does this classifier work? What main parameters can be specified for this classifier?
4. Under "Test options" select "Use training set" and under "More options" check "Output predictions". Now click "Start" to start training the model. You should see a stream of output appear in the window named "Classifier output". What do each of the following sections tell you about the model?
   (a) "Predictions on ..."
   (b) "Summary"
   (c) "Detailed accuracy by class"
   (d) "Confusion matrix"

5. Under "Results list" you should see your model, right click on it and select "Visualise classifier errors", points marked with a square are errors i.e. incorrectly classified. How do you think the classifier performed on the test data?

6. Under "Test options" vary the option selected i.e. "cross-validation" or "percentage" and their parameters i.e. "folds" and "%". Then start the training phase again for each model. For each model analyse the classifier output and visualise the classifier errors. How do the different training techniques affect the model? Which technique performed the best? How does this compare to your initial prediction in 4?

7. Repeat the exercise 6 with the "J48" (Decision Tree) and "RBFNetwork" classifiers. How do these compare to each other? How do these compare to the "MultilayerPerceptron"?