ECS607/766 Data Mining 2017/18

Assignment 3

Introduction

The outcome of this lab is to get experience with classification problems, and the concepts of priors, dimensionality, naïve bayes and decision trees. This lab will use both matlab and weka. Start by downloading the zip file of lab materials from qmplus. Questions in ***red*** are assessed toward your final grade. You **MUST** both submit your answers online on QM-Plus and show the submitted version to one of the Demonstrators for marking.

1. Decision Trees

1. Open weka and load the soybean dataset.
    - This is about predicting what disease a soya plant has depending on various observations of the plant. The 36th attribute is the target variable (disease type).
2. Let's train a decision tree.
    - Classify => Choose => Tree => J48 to get a decision tree learner.
    - Click "Percentage split" 66%. And then Start.
    - You can see the overall accuracy (Correctly classified instances), and the confusion matrix about which diseases were mixed up with each other.
    - Take note of: Number of Leaves, Size of Tree, and Test Accuracy.
    - The default setting for J48 trees in weka is Pruned.
    - Now switch off tree pruning by clicking the string next to Choose and set unpruned => true.
    - Take note of the new: Number of Leaves, Size of Tree, and Test Accuracy.
    - ***How and why are these numbers related in the pruned and unpruned cases?*** [1 mark]

2. Data Proportions with MaxEnt + Naïve Bayes

1. Invoke matlab. Edit the file lab4_1.m.
    - Recall that you can run matlab code one cell at a time by ctrl-enter. You can also set breakpoints by clicking to the right of the line number and step through. You can see a variable in a running program by typing the name, clicking the workspace browser, or mouse-over the variable name.
2. Run Cell 1.
    - This first loads the Iris flower dataset http://en.wikipedia.org/wiki/Iris_flower_data_set. This is a classic dataset about identifying the types of flowers based on some measurements of their petals. The full data has four attribute columns ('**size meas**'), but we will work with the first two ('**size Xtr**'), plotted in Fig 1.
3. Run cell 2. This uses the logistic regression classifier from last week to learn a flower classifier.
    - Note the accuracy.
    - Note how many instances are in each class? (Ytr is the class variable of training data. Hint: h=hist(Ytr,1:3))

4. Cell 3. Now we assume that the classifier is trained in some country A, where the flower proportions are different (flower type 3 is 5x less common)
    - Note the train and test accuracy and the new decision boundary plot.
    - The routine **confusionmat** generates a matrix showing how each category is confused with each other category. (True class on rows, estimated on columns)
    - Try show the confusion matrix of true versus estimated classes **cmat=confusionmat `(YtrA,predTrA)`**
    - Although the overall accuracy is high, one class is being mis-classified dramatically (try the normalized conusion matrix **`normalise(cmat,2)`**)
    - This could be a problem if category #3 is important to detect in this a application domain.
    - Why has this happened?
    - **What accuracy is the minority class detected with for the test data, and what other class is it confused with?** [1 mark]

5. Cell 4. Now this classifier is taken to another country B, where the flower proportions are different again (flower type 2 is 5x less common).
    - Compare the flower frequencies with **h=hist(YtrA,1:3)** and **h=hist(YtrB,1:3)**
    - Note the classification accuracy? What is the confusion matrix between class 2 & 3? Note the off strong diagonal elements.
    - Note the distribution of blue and green points versus the decision boundaries in the plot. (If it is working well, green points should be in the red space, blue points in the yellow space).
    - **What is the new accuracy for class 3? What happened here?** [1 mark]

6. Cell 5. Now switch from MaxEnt/LR to Naïve Bayes classifier back in country A.
    - Note the performance and confusion matrix output.
7. Cell 6. Adjusting the prior.
    - Naïve Bayes classifier can independently specify the prior. Recall from the class that it just learns the likelihood. At test time the prior is combined with the likelihood via Bayes theorem to actually classify.
    - By default it uses the observed data frequency to estimate the prior. But you can specify yourself. Try specifying a balanced prior even though the data is imbalanced. (`S.prob = [1/3,1/3,1/3]; S.group = [1, 2, 3];`
    - `nb = NaiveBayes.fit(XtrA, YtrA, 'Prior', S);`
    - Now when you run the classifier, observe the movement in the decision boundaries and the change in the confusion matrix between Cell 5 &6.
8. Cell 7. Now we can effectively apply the Country A classifier to country B – without retraining. Instead we simply update the prior to reflect the flower frequency in country B.
    - Now we can apply the NB classifier learned in country A to country B by adjusting the prior. (Try **S.prob = [x,y,z]** .)
    - **What prior should you set to get maximum accuracy in country B? What accuracy do you get by setting this?** [1 mark] (You should be able to get noticeably more accuracy than the direct use of LR classifier from A to B in cell 4. Hint: A good prior to use reflects the relative frequency of flowers in country B).

3. Scaling with # of dimensions (Aka Attributes, Features)

1. Edit the file lab4_2.m. Cell1.
    - I create some data with 25 examples in 2 classes. As you can see in Figure 1, the data is only very weakly separable. So it is a tough problem.
2. Cell 2: Run logistic regression.
    - Note the train and test accuracy.
3. Cell 3: Run naïve Bayes.
    - Note the train and test accuracy. They should be similar.
4. Now lets suppose instead of only 2 attributes, there are 200 attributes, each of which are weakly informative, (set dim = 200 in cell 1 and re-run everything).
    - Note that LR takes longer to run.
    - Note the train and test accuracy of each approach
    - ***What happened here?*** [2 marks]

4. Exploring ROC curves

1. Q2 revealed the nuances of classification where there are varying amounts of data imbalance, and highlighted that the challenge of classification where one particular class is of more interest than the others. Especially if it's the minority class its likely to be misclassified.
2. One way to address this is to vary the threshold of the classifier: How strongly should a particular class be indicated before making a decision to detect it.
3. Edit lab4_3.m. Run cells 1,2,3. This generates the same data as for Q3, and trains LR and NB models to classify it.
4. Observe that Cell 4 is now setup to change the decision threshold instead of assuming its 0.5. Try so me different threshold numbers and observe how the confusion matrix and accuracy changes. By changing the threshold number you can make the classifier prefer class 1 or class 2, which could be useful in an application where they have varying importance.
5. **Bonus Question:** Cell 5 also computes true positive rate (TPR) and false positive rate (FPR) for the classifiers. Add a loop over thresholds and try to make a ROC curve to compare the classifiers over all thresholds. Once you compute a vector of TPR and FPR, you can try to plot them against each other (e.g., **plot(FPR,TPR)** to show the ROC curve).

Submission

Remember to show your work to a TA as well as upload to qmplus by the deadline.