ECS607/766 Data Mining 2015/16

Lab 4: Classification 2

Introduction

This lab covers the concepts of **priors**, **dimensionality**, **Bayes** and **trees** in **classification problems**.

We will use both Matlab and Weka. Start by downloading the zip file containing the lab materials from QMplus and remember that questions in ***red*** will be assessed toward your final grade.

1. Trees

1. Open Weka and load the soybean dataset.

    - This dataset will be used to predict what disease a soya plant has based on a number of observations of the plant. The 36th attribute *class* is the target variable, i.e. the disease type.

2. Let's train a so-called J48 decision tree. Its default setting is Pruned, and in this exercise we will study the effects on pruning on the resulting tree.

    - *Classify* => *Choose* => *Tree* => *J48* to select a decision tree learner.
    - Click *Percentage split* and set it to 66% and then click *Start*.
    - You can see the overall accuracy (*Correctly classified instances*) and also a confusion matrix, which identifies the number of cases that have been wrongly labelled.
    - Take note of *Number of Leaves*, *Size of Tree*, and *Test Accuracy*.

    - Now switch off tree pruning by clicking the text box next to *Choose* and set *unpruned* => *true*.
    - Take note of the new *Number of Leaves*, *Size of Tree*, and *Test Accuracy*.

    - **Compare the *Number of Leaves* and *Size of Tree* of both trees (i.e. with and without pruning) and explain any differences observed.**

    - **Compare the *Test Accuracy* of both trees. Which tree shows a better performance? Explain your observation based on the principles of tree pruning.**

2. Data Proportions with MaxEnt + Bayes

1. Invoke Matlab and 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 any variable in a running program by typing the name in the command window, using the workspace browser, or hovering the mouse pointer over the variable name.*

2. Run Cell 1.

    - This first cell loads the Iris flower dataset. The Iris flower fataset is a classic dataset used to identify types of flowers based on features describing their petals. The full dataset contains four attribute columns (type ***size(meas)*** in the command window), but we will only work with the first two (type ***size(Xtr)***). Figure 1 represents the resulting dataset.

3. Run cell 2. This uses a logistic regression classifier to create a flower classifier.

    - What is the accuracy?
    - How many instances are in each class? (Hint: Note that *Ytr* is the vector containing the labels or desired response for each sample in the training dataset)

4. Run Cell 3. We now assume that the classifier is trained in some region *A*, where the proportions of each type of flowers are different. Specifically, we assume that flower type 3 is 5 times less common.

    - Note the train and test accuracy and analyse the new decision boundary plot.
    - The routine ***confusionmat*** generates a confusion matrix that shows how each category is confused with each other category. The true classes correspond to the rows of the confusion matrix, whereas the predicted classes correspond to the columns.
    - Obtain the confusion matrix corresponding to the training dataset by typing ***cmat=confusionmat (YtrA,predTrA).***
    - Note that although the overall accuracy is high, one class is being heavily mis-classified. By normalizing the confusion matrix ***cmat*** this observation will become clearer (type ***normalise(cmat,2)***).
    - **Which class is being heavily mis-classified? Why has this happened?**
    - **Obtain the test detection accuracy for this class and identify the other class that this class is being confused with?**

5. Run Cell 4. Now let's assume our classifier is taken to another region B, where the flower proportions are different again, specifically flower type 2 is 5 times less common.

    - Compare the flower frequencies by typing **h=hist(YtrA,1:3)** and **h=hist(YtrB,1:3)** in the command window.

- Note the classification accuracy and obtain the confusion matrix.
- Analyse the distribution of blue and green points in the predictor space and the resulting decision boundaries (note that if samples were correctly classified, the green samples should be in the red space, and the blue points in the yellow space).
- **What is the new accuracy for class 3? Compare this accuracy with the accuracy obtained in the previous section and explain any discrepancies.**

6. Run Cell 5. In this cell we obtain a Naïve Bayes classifier from a dataset corresponding to the region A.

    - Note the classifier's performance and the confusion matrix output.

7. Cell 6. Adjusting the priors.

    - In Naïve Bayes classifiers, priors can be specified. At test time the prior is combined with the likelihood via Bayes theorem to classify the samples.
    - By default, Bayes classifiers use the observed data frequency to estimate the prior, although you can specify them yourself.
    - Specify a balanced prior (even though the data is imbalanced) by typing `S.prob = [1/3,1/3,1/3]; S.group = [1, 2, 3];` on the command window.
    - Then run the Bayes classifier using the new priors by typing `nb = NaiveBayes.fit(XtrA, YtrA, 'Prior', S);`
    - Compare your results against the ones obtained by running Cell 5, by analyzing any changes in the decision boundaries and in the confusion matrix.

8. Run Cell 7. Now we can effectively apply the region A classifier to reiong B without having to retrain it. Instead of retraining the classifier, we will simply update the priors to reflect the flower frequency in region B.
    - Adjust the priors (try ***S.prob = [x,y,z]***).
    - **What prior should you use to get maximum accuracy in region B? What accuracy do you get by using this value?** Note that you should be able to get more accuracy than by using of the MaxEnt classifier from region A in B (Cell 4)and remember that a good prior should reflect the relative frequency of flowers in region B.

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

1. Edit the file lab4_2.m.
2. Run Cell1:
    - You will create a dataset consisting of 25 examples in 2 classes. As you can see in Figure 1, the dataset is weakly separable.
3. Cell 2: Run logistic regression.
    - Note the train and test accuracy.
4. Cell 3: Run naïve Bayes.
    - Note the train and test accuracy. They should be similar.

5. Now lets suppose that instead of 2 attributes, there are 200 attributes, each of which are weakly informative.
    - Set *dim* = 200 in Cell 1 and re-run the remaining cells.
    - Note that LR takes longer to run.
    - Note the train and test accuracy of each approach
    - **Compare the performance of both classifiers in the 2-class scenario with the performance in the 200-class scenario and explain any differences you might observe.**

4. Exploring ROC curves

1. In question 2 we worked with imbalanced datasets. Specifically, we observed that minority classes are likely to be misclassified. One way to address this is to adjust the boundary of the classifier.
2. Edit lab4_3.m and run cells 1,2,3. By doing so, you will generate the same data that we used in question 3 and will train maxEnt and Bayes models to classify it.
3. Observe that Cell 4 is now setup to change the decision boundary by adjusting a threshold that by default is 0.5. Try different values for the threshold and observe how the confusion matrix and accuracy change. By changing the threshold value, you can make the classifier prefer class 1 over class 2, which could be useful in an application where they have varying importance.
4. **Bonus Question:** Cell 5 also computes true positive rate (TPR) and false positive rate (FPR) for the classifiers. Add a loop over the thresholds and try to build a ROC curve to compare the classifiers over all the 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 upload your report to QMplus by the deadline!