ECS607/766 Data Mining 2017/18

Assignment 5: Metrics and Clustering

Introduction

The goal of this lab is: (i) to go beyond accuracy and confusion matrix metrics you are already familiar with for supervised learning of classifiers, and appreciate the role of ROC curves, and (ii) to get experience with unsupervised clustering using K-means. The lab will use matlab. Start by downloading the zip file of lab materials from qmplus. Questions in ***red*** are assessed toward your final grade. You **MUST** upload your sheet to qmplus and get it checked by a TA by the end of the next weeks session.

1. Kmeans

In this exercise, we will explore the code of K-means clustering, evaluation of clustering quality, and the relation between clustering and classification.

1. Invoke matlab. Load lab6_1.m.
    - This contains some skeleton of code for K-means.
2. Run Cell #1
    - This loads the iris flower dataset that you are already familiar with. You should see a window with 2 of the 4 dimensions of this data (flower petal geometry) colored by types of flower. Today, instead of learning to classifying it, we will cluster it.
3. Run Cell #2.
    - This cell initializes K-means. You should see a figure showing the flower datapoints, and some randomly initialized cluster centres. Note that as this is unsupervised clustering, we don't assume any knowledge of the true labels.
    - Recall that in K-means, data points will be assigned to the cluster centre that they are closest to.
    - How do you imagine the data points shown will be classified based on similarity to the illustrated cluster centres?
4. Run Cell #3.
    - This does the first 'E-step' of K-means. You should see the points now assigned to nearest clusters. The second return value gives the summed distance of every point to its nearest cluster.
5. Go to Cell #4.
    - The next step in K-means is to update the cluster centers so they are in the middle (mean of) they points that have been assigned to them. It's a good exercise to try and program this yourself. Or you can use the line `centres(c,:) = mean(X(lab==c,:),1);` as a shortcut.
    - Run the completed cell 4. This does the first 'M-step' of K-means. You should see the clusters have been moved to the mean of their data.
    - Compare the value of totalDistToCluster from Cell #3 and totalDistToClusterUpdated from Cell #4. Did the overall clustering quality improve (total distance down)?
6. Load lab6_2.m.

- Run the whole script, and observe the Figure 3 plot of cluster quality continually improving.
- Repeat the experiment varying the value of K in cell 2 from 2 to 5.
- What do you observe about the dependence of the final cluster quality on the number of clusters K used? [1/2 mark]
- Why?
- Set K back to 3. Find the Cell #2 line **rng(0);**
- This is a random seed that will pick different initial conditions for the cluster. Try different parameters (positive integers) for rng(XXX).
- What do you observe?
- Find a seed that gives you a cluster quality of 62.8 or better. [1/2 mark]
- Try to change the definition of distance used for clustering. IN the call assignPointsToCluster(X,centres,'euclidean'); try distance types 'cityblock' or 'cosine'. What do you observe about difference in the resulting clustering??
- **Bonus:** The current code runs for a fixed number of iterations, rather than until convergence. Change the loop in Cell 3 so that it runs to convergence. How many iterations does it take to converge?

7. Load lab6_3.m
   - The goal of clustering is to discover clusters (classes) in the data: In the context of unsupervised learning where pre-labeled examples are not known in advance.
   - Run the whole script. The final cell here compares every possible assignment of clusters to true class labels. To evaluate the match between grouping of data points into clusters and the true labels. What clustering accuracy do you get? (Notice the difference between accuracy, where more is better. And quality, where lower distance is better)
   - Find the commented Cell 1 line **X=meas; d=4;** . Uncomment it, and then re-run the script. What's happening here?
   - In the cluster display windows, you should see that the cluster boundaries shown are no longer so clean as before. The colors indicating cluster are mixed up. Why is this?
   - Has the clustering accuracy improved from before to after of the uncommenting above? Why? [1 mark]
   - **Bonus:** Edit the visualization to use the function **plot3** to visualize three of the K-means clustering dimensions instead of just 2.

2. ROC Curves

In this exercise, we will explore the ROC curves as a means to evaluate classifiers, and the benefits they bring compared to simple accuracy metrics. This exercise is about material in the Classification 2 lecture.

1. Load lab6_4.m.
2. Run the first cell. It sets up a two-class classification problem based on the iris flower data you are already familiar with. As you can see the data is quite overlapped, so it will be hard to classify accurately.
3. Run Cell2A and Cell2B. They train Naïve Bayes and LR classifiers respectively. Observe the decision boundaries in the figures.
4. Typically we make decisions based on which class has greater probability (i.e., which class has > 0.5 probability in the binary case).
5. Sometimes its more important to detect one class than the other. In this case we can tune the threshold to prefer one class or other. In Cell 3, you can set the threshold value. Try some threshold values between 0 and 1. Observe the visualized decision boundaries in Fig 1 and 2 and the displayed confusion matrix.
6. The threshold to use provides a parameter for the user of a system to control. In this way they can for example maintain a maximum false positive rate, or a minimum true positive rate according to the application. One way to evaluate a classifier so its suitability for any application can be checked is to calculate true positive rate (TPR) and false positive rate (FPR) for the entire range of thresholds.
7. Cell four contains the inner content of an loop to compute TPR and FPR. Add the missing loop required to try a bunch of thresholds and fill in the TPR and FPR vectors. [1 mark]
8. Plot tpr (y-axis) against fpr (x-axis) to visualize the resulting ROC curve. [1 mark]
9. From the ROC curve, you can see that one classifier or the other may be slightly preferable depending on the required TPR/FPR constraints of the particular application.
10. The Area under the ROC curve is the preferred metric to compare two curves independently of a specific choice of threshold. Compare the AUROCs of the two classifiers with **[~,~,~,aucLR]=perfcurve(Yte,pTeLR,1)** and **[~,~,~,aucNB]=perfcurve(Yte,pTeNB(:,2),1)**
11. Which one is preferable by AUC metric? [1 mark]
12. Suppose for a particular application, the maximum allowed FPR is 0.16. Which classifier is preferable (obtains the maximum TPR given this FPR constraint)? [1 mark]