Computer Science 315
Assignment 3

2016

For this assignment, you will first implement a nearest centroid classifier to serve as a baseline, and then use the `scikit-learn` implementation of the logistic regression classifier. (You can find the API for this at http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.) Finally, you will complete and use a logistic regression classifier based on the skeleton code provided in your resources directory.

Getting started

Use the provided wine data set and do the following.

1. Fit an LDA model to the training data, and transform it and the test data, using two components.

2. Implement a 1-nearest prototype classifier to classify the wine data. A nearest prototype classifier is a classifier that makes predictions by storing labelled prototype data points, and labelling new points by considering the labels of nearby prototypes. For a 1-nearest prototype classifier, the label of the nearest prototype is used as the classification result. Your function should take as input data values to be assigned, the prototype feature vectors, and the corresponding class labels for these prototypes. It should return the assigned class label for each data value. It is easy but inefficient do to the classification using `for` loops; it is recommended you use broadcasting instead, in which case your code should only be about four lines long. If you use broadcasting you may include the code as part of your report for bonus points, otherwise do not include it. Use this 1-nearest prototype classifier to classify the transformed training and test data when using a single prototype for each class, located at the class means of the transformed training data. (When these prototypes are used, we refer to the classifier as a *nearest centroid classifier*.) Print a confusion matrix showing your classification results.

3. Fit the `scikit-learn` logistic regression model to the transformed training data. Classify all the transformed test observations using the logistic regression model learned from the training data. Print a confusion matrix of the results.

4. Use the the full wine training data set (not transformed to lower dimensional space) to fit the `scikit-learn` logistic regression model, and classify all the observations in the test data set. Print a confusion matrix. How does it compare to the transformed results obtained above?

Decision boundaries

We now want to investigate the decision boundaries. In order to be able to draw the decision boundaries on a graph, transform the wine data to two dimensions using LDA. Needless to say, you should build your models using the training set.

I. Binary Classification

1. Use LDA to project the wine training data to two dimensions. Display the three classes and decide which two classes are the most difficult, i.e. show the most overlap. Use these two classes to train your logistic regression classifier.
2. Classify the observations in the training and test sets using `scikit-learn`. Display the results on two graphs for each set. In the first one, color code the actual two classes. In the second graph, color code the two predicted classes.
3. Extract the parameters of the decision boundary from the `scikit-learn` class, and draw it together with the color-coded predicted classes. You should be able to tell whether the decision boundary makes sense. It is important that you draw the decision boundaries using the parameters of the classifier.
4. Using the skeleton code provided, develop your own (binary) logistic regression model. You have to code the gradient and Hessian used in the training of the model. Use your implementation to repeat the binary classification tasks described above and compare your results with those obtained using the `scikit-learn` implementation.

II. Multi-class classification

1. Visualize the performance of the nearest centroid algorithm by plotting the actual labels for the two-component transformed test data. as well as the decision boundaries. In this case it may be easier to actually classify all the values in a grid over the entire observation region. You may find the following resource useful for the display: `http://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html`.
2. Do the same with the three-class logistic regression classification using `scikit-learn`.