CS 562: Problem Set 3 (Part 1)

March 4, 2014

Instructions

These problems are to be started and worked on in class in groups of two (or inidividually). Please make sure that you have ready access to a computer - either the clamshell server or your own laptop running Linux-based sofware with all the required Python software. The warmup problems do not have to be submitted: they are simply meant to get you familair with pieces of code that we will use elsewhere.

Warmups

1. Study and run the following programs from the scikit-learn documentation: all the programs have the common base URL **http://scikit-learn.org/stable/auto_examples/**

   (a) plot_random_dataset.py at datasets/plot_random_dataset.html

   (b) plot_ols.py at linear_model/plot_ols.html

   (c) plot_iris.py at svm/plot_iris.html

   (d) svm_gui.py at applications/svm_gui.html

   (e) plot_pca_iris.py at decomposition/plot_pca_iris.html

   (f) plot_svm_margin.py at svm/plot_svm_margin.html

   (g) plot_pca_3d.py at decomposition/plot_pca_3d.html

   (h) plot_sgd_separating_hyperplane.py at linear_model/plot_sgd_separating_hyperplane.html

2. Under Resources→Data on Sakai, you will find a dataset archived as **mushrooms.tar**: it includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one.

There are 8124 instances in all with 23 nominally valued attributes which are described in the README.txt file. The 0^{th} attribute indicates the target label: the mushroom instance is classified as poisonous (p) or edible (e).

Use the **pandas** package to read the data instances into a **DataFrame** object. These are tables that are generalizations of numpy arrays that can store arbitrary Python datatypes with the same fast access to any index. Use the DataFrame methods - which are similar to numpy array methods - to convert the nominal attributes (string-valued) to numeric values.

Problems

1. Example 12.9 in the MMDS textbook describes in detail the initial iterations in the SVM classification of the dataset with points (1,4), (3,4) and (2,2) labeled +1 and (1,1), (2,1) and (3,1) labeled -1. Write a program that computes the weights as described in the example: this is the standard SVM gradient descent technique. Your program should be able to pause during each iteration so that in the iteractive ipython shell, you will be able to see how the weights and the predictions are changing, and also which points are misclassified. The program should be designed so that it quits under user control.

2. Exercise 12.3.3 from MMDS: The following training set obeys the rule that the positive examples all have vectors whose components have an odd sum, while the sum is even for the negative examples.

   ([1,2], +1) ([3,4], +1) ([5,2], +1)
   ([2,4], −1) ([3,1], −1) ([7,3], −1)

   (a) Suggest a starting vector w and constant b that classifies at least three of the points correctly.

   (b) Starting with your answer to (a), use gradient descent to find the optimum w and b. You can either use the program developed in Problem 1 to do so, or use the svm package in scikit-learn.

3. Use the first 5000 instances of the mushrooms dataset (after converting the features to numeric values) to train a **linear regression** model using the **sklearn.linear_model** package and determine how well the trained model predicts the class label of a random collection of 25 instances from among the remaining 3124 instances. Summarize and visualize your results appropriately.

4. Use numpy to compute the eigenvalues and eigenvectors of the matrix

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 5 \end{matrix}$$

   Now, implement the matrix powers method and compare results.

5. Use PCA on the Boston house prices dataset from scikit-learn. Summarize and visualize your results appropriately. Remember that you should first do mean normalization and feature scaling so that all the features have similar scales, i.e. replace each feature x with $\frac{x-\mu_x}{\sigma_x}$).