Artificial Intelligence

COMS 4701 Section 1 – Summer 2016

Homework N°3: Machine Learning

Due Sunday June 26th, 2016 @11:55pm

Overall Instructions

1. Please use Python 2 for this homework. You may use Python packages to help you solve the problems and plot the results, including **numpy**, **matplotlib**, and **scikit-learn**.
2. Please **comment all important sections** of your code.
3. Your submission package must be a zipped file that includes the following:
    (a) a **single PDF** file containing all your plots, responses, and comments. Please keep your writing as **concise** as possible.
    (b) code for experimentations and plot generation named by problem (i.e. problem#.py)
    (c) (optional) a README.txt file to document your code
4. Name your submission as firstname_lastname_uni.zip and submit on Canvas.

Problem 1: Linear Regression

In this problem, you will work on linear regression with multiple features using gradient descent. We will use the dataset **girls_age_weight_height_2_8.csv** (derived from CDC growthchart data).

1. **Data Preparation and Normalization:** Once you load your dataset, explore the content to identify each feature. Remember to add the vector 1 (intercept) ahead of your data matrix.
    a) You will notice that the features are not on the same scale. They represent age (years), and weight (kilograms). Print the mean and standard deviation of each feature in your data. The last column is the label and represent the height (meters).
    b) Scale each feature by its standard deviations and set its mean to zero. You do not need to scale the intercept. For the each feature x (a column in your data matrix), use the following formula:

$$x_{\text{scaled}} = \frac{x - \mu(x)}{stdev(x)}$$

2. **Gradient Descent:** Implement gradient descent to find a regression model. Initialize your β's to zero. Recall the empirical risk and gradient descent rule as follows:

$$R(\beta) = \frac{1}{2n}\sum_{i=0}^{n}(f(x_i) - y_i)^2$$

$$\forall j \qquad \beta_j := \beta_j - \alpha\frac{1}{n}\sum_{i=0}^{n}(f(x_i) - y_i)x_i$$

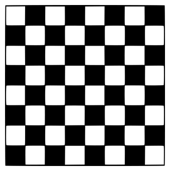
    a) Run the gradient descent algorithm and plot the risk function with respect to different learning rates. Use learning rates $\alpha \in \{0.005, 0.001, 0.05, 0.1, 0.5, 1\}$ and run your algorithm for 50 iterations.

b) Compare the convergence rate when α is small versus large.

c) Which α is best? Use this α to run gradient descent and report the β's.

d) Use the β vector from part c to make a height prediction for a 5-year old girl weighting 20 kilos (don't forget to scale!).

Problem 2: Classification



In this problem you will use the support vector classifiers in the sklearn package to learn a classification model for a chessboard-like dataset.

1. Open the dataset chessboard.csv in python. Make a scatter plot of the dataset showing the two classes with two different patterns.
2. Use SVM with different kernels to build a classifier. Make sure you split your data into training (60%) and testing (40%). Also make sure you use a stratified sampling (i.e. same ratio positive to negative in the training and testing datasets). Use **cross validation** this time instead of a validation set (train-test splitting and cross validation are both functionalities that are readily available in Sklearn).
   (a) Show the performance of the SVM for linear, polynomial and RBF kernels with different settings of the polynomial order, sigma and C value. Search for a good setting of these parameters to obtain high classification accuracy. Report all your results on crossvalidation and test data.
   (b) Plot the decision boundary for the best parameters for each kernel.
3. Same procedures as in part 2, use logistic regression instead.
4. Same procedures as in part 2, use decision trees instead.
5. In your write up, discuss which classification method is better suited for this dataset and why.

Problem 3: Clustering for Image Segmentation

In this problem you will experiment with two clustering algorithms to classify pixels in a given image into different groups according to their RGB values. You can read about the applications of image segmentation here: `https://en.wikipedia.org/wiki/Image_segmentation`. The image files you will use for this assignment is **trees.png**.

1. **K-means:** For this part you will experiment with the k-means algorithm. Choose 3 representative k values and use the k-means algorithm in sklearn to process trees.png. Report and comment on your results. Below is an example of trees.png with k=3.

2. **Spectral Clustering (Bonus Question):** Choose a dataset on which spectral clustering works better than k-means. You may use the spectral clustering algorithm implemented in sklearn for this problem. Compare the results when you apply both methods and explain your results.