HW 1

CS6510: Applied Machine Learning
IIT-Hyderabad
Aug-Nov 2016

Max Points: 60
Due: 26th Aug 2016 11:59 pm

Instructions

- Please use Google Classroom to upload your submission by the deadline mentioned above. Your submission should comprise of a single ZIP file with all your solutions, including code.
- For late submissions, 10% is deducted for each day (including weekend) late after an assignment is due. Note that each student begins the course with 5 grace days for late submission of assignments. Late submissions will automatically use your grace days balance, if you have any left. You can see your balance on the CS6510 Marks and Grace Days document under the course Google drive.
- You should use PYTHON for the programming assignments.
- Please read the department plagiarism policy. Do not engage in any form of cheating - strict penalties will be imposed for both givers and takers. Please talk to instructor or TA if you have concerns.

1 Theory [30 points]

1. **(6 points)** An urn contains 80 blue balls and 60 white balls. An observer samples *with replacement* 100 balls from this urn. Assume

that A is the number of white balls in first 60 draws and B be the number of white balls in the next 40 draws. Further, C = A + B is the total number of white balls in the sample.

- What are the distributions of A, B and C? [*2 points*]
- Are A and B dependent or independent? [*1 point*]
- Find P(A = s|C = q) for $0 \leq s \leq q$,where q is any fixed number between 0 and 100. [*2 points*]
- Are A and C independent? [*1 point*]

2. **(5 points)** A random sample $\{Y_1, \ldots, Y_n\}$ follows the PDF given by:

$$f_Y(y) = \frac{y}{b^2} e^{\frac{-y^2}{2b^2}}$$

with $y > 0$ and a user-defined parameter b.

(a) Show that $f_Y(y)$ is a valid PDF. [*1 point*]

(b) Obtain the likelihood and log-likelihood function for Y in terms of parameter b. [*2 points*]

(c) Find the maximum likelihood estimate for parameter b. [*2 points*]

3. **(7 points)**

(a) Prove or disprove: Empty set is a vector Space. [*1 point*]

(b) Show that the inverse of $M = I + (\mathbf{u}\mathbf{v}^T)$ is of the type $I + \alpha(\mathbf{u}\mathbf{v}^T)$, where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, $\mathbf{v}^T\mathbf{u} \neq 0$. [*2 points*]

(c) Continuing from part (b), find α. [*2 points*]

(d) (contd.) For what $\mathbf{u}$ and $\mathbf{v}$ is M singular? [*1 point*]

(e) (contd.) Find the Null space of M, if it is singular. [*1 point*]

4. **(4 points)**

(a) Consider the 2×2 matrix:

$$A = \begin{bmatrix} -2 & 2 \\ -6 & 5 \end{bmatrix}$$

There exists vectors such that when the matrix A acts on those vectors, the subspace of the vectors does not change. Mathematically, $Ax = \lambda x$. The vectors x are called *eigenvectors* and the values λ are the corresponding *eigenvalues*. Find the eigenvalues and corresponding eigenvectors for the matrix A. [*2 points*]

(b) (contd.) Consider a diagonal matrix Λ which has the eigenvalues of A as its diagonal entries. Find the matrix U such that the equation $AU = U\Lambda$ holds. [*1 point*]

(c) Note that we can write the matrix A as $A = U\Lambda U^{-1}$. Find the inverse of the matrix U computed in part (b) and verify. [*1 point*]

5. **(8 points)** In k-nearest neighbors (k-NN), the classification is achieved by majority vote in the vicinity of data. Given n points, imagine two classes of data each of $n/2$ points, which are overlapped to some extent in a 2-dimensional space.

   (a) Describe what happens to the training error (using all available data) when the neighbor size k varies from n to 1. [*2 points*]

   (b) Predict and explain with a sketch how the generalization error (e.g. holding out some data for testing) would change when k varies? Explain your reasoning. [*2 points*]

   (c) Propose a method to determine an appropriate value for k. [*1 point*]

   (d) Give two reasons (with sound justification) why k-NN may be undesirable when the input dimension is high. [*3 points*]

2 Programming (30 points)

1. **(6 points)** The following questions should be done using `numpy` operations. Use of *iterative constructs* is not allowed.

   (a) Create an $n \times n$ array with checkerboard pattern of zeros and ones. [*2 points*]

   (b) Given an $n \times n$ array, sort the rows of array according to m^{th} column of array. [*2 points*]

   (c) Create an $n \times n$ array with $(i+j)^{th}$-entry equal to $i+j$. [*2 points*]

2. **k-NN for Breast-cancer Diagnosis (9 points)**
   This link contains the breast cancer database. Please refer to this link for the details of what each attribute represents in the data. In short, Class **2** represents benign tumor and Class **4** represents malignant tumor. Solve the given classification problem using k-NN classifier in the following ways: (Use random 80-20 split of the data into train and test respectively).

   (a) Using `KNeighborsClassifier()` function of `scikit-learn` library. [*3 points*]

   (b) Write your own custom implementation of K-Nearest Neighbors without using any functions of `scikit-learn` (You are free to use `numpy`). [*3 points*]

   (c) Compare the *accuracy* and *running time* for your solutions in parts (a) and (b). Do you observe any differences? If so, please explain your observations.What are the ways in which the slower one can be made faster? [*3 points*]

   Note: ? in the dataset denotes missing attributes. Please mention clearly (in your submission report) how you handled the missing attributes in data.

3. **Decision Trees for Oropharynx Cancer (OPC) Radiomics Challenge (15 points)**
   Visit this link to access this Kaggle challenge dataset. `training.csv` will provide details of what each attribute represents in the data (you can also read the README file on the above link). Note that the last column in this comma-delimited file is the *classification attribute*, and will always contain the values 0 or 1. Use the first 140 samples from `training.csv` as training data and the remaining 10 samples as the test data (do not consider its output while training your method).

   (a) Implement the decision tree classifier (ID3) to solve this problem using `scikit-learn` (you can use `DecisionTreeClassifier()`). Submit your code as well as a writeup in your report showing the attribute selected and the corresponding information gain at each split when running your decision tree learning algorithm. [*4 points*]

(b) Plot a line graph with number of leaf nodes on the x-axis and classification accuracy on the y-axis. What does this tell you? [*2 points*]

(c) Pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting. One of the well-known pruning methods is *chi-square pruning*. Include in your report a second decision tree obtained by performing chi-squared pruning on the first decision tree. You can use this link as reference. Did pruning help? [*3 points*]

(d) Information gain is one approach to make an optimal split, but this is not the only one. Here we will consider a criterion where we try to minimize the weighted misclassication rate. Given 140 training samples of the form: $\{\vec{x}^{(i)}, y^i\}$, where $\vec{x}^{(i)}$ is the feature vector for the corresponding sample i and y^i is the lable of one sample i. Let, $\vec{x}(j)^{(i)}$ denote the j^{th} attribute of the feature data point i. To pick a new split criterion, we pick a feature attribute, a and a threshold value t, to use. Now, consider:

$$p_{below}(a,t) = \frac{1}{140}\sum_{i=1}^{140}\mathbb{I}(\vec{x}(a)^{(i)} \leq t)$$

$$p_{above}(a,t) = \frac{1}{140}\sum_{i=1}^{140}\mathbb{I}(\vec{x}(a)^{(i)} > t)$$

where $\mathbb{I}$ is the indicator function, which returns 1 when true. Let:

$$l_{below}(a,t) = \text{Mode}(\{y_i\}_{i:x(a)^{(i)} \leq t})$$

$$l_{above}(a,t) = \text{Mode}(\{y_i\}_{i:x(a)^{(i)} > t})$$

The split that minimizes the weighted misclassication rate is then the one which minimizes:

$$O(a,t) = p_{above}(a,t)\sum_{i:x(a)^{(i)}>t}\mathbb{I}(y^{(i)} \neq l_{above}(a,t))$$

$$+p_{below}(a,t) \sum_{i:x(a)^{(i)} \leq t} \mathbb{I}(y^i \neq l_{below}(a,t))$$

Please modify your code to perform splits according to this criterion. [*6 points*]

NOTE: Submit your Python file in such a way that it can be run using the following command: `python decisiontree.py training.csv`.