Introduction to Machine Learning

Assignment 2

Submission date:25/02/2020 , 23:59

General Instructions:

- The solution should be formatted as a report and running code should be included in a digital form.
- The solution can be done in pairs independently to other groups. Identical (or very similar solutions) are not allowed!
- You may choose whether to write your code in Python or Matlab.
- Please submit the assignment via Moodle.
- The code must be reasonably documented.

Question 1: Regression

Use a regression method to predict housing prices in suburbs of Boston.

Data: You'll find the data in the file "housing.data". Information about the data, including the column interpretation can be found in the file "housing.names". These files are part of UCI Machine Learning Repository and can be downloaded from https://archive.ics.uci.edu/ml/machine-learning-databases/housing/

a. Predict the median house value (the 14th, and last, column of the data) based on the other columns. Use a linear regression model to predict the house values, using squared error as the criterion to minimize:

$$y = f(x, \hat{w}) = \hat{w}_0 + \sum_{i=1}^{13} \hat{w}_i x_i ,$$

$$\text{where } \hat{w} = \arg\min_w \sum_{t=1}^{n} (y_t - f(x_t, w))^2 ,$$

here y_t are the house values, x_t are input vectors, and n is the number of training examples.

Write the following functions:

- A function that takes as input weights w and a set of input vectors $\{x_t\}_{t=1}^n$ and returns the predicted output values $\{y_t\}_{t=1}^n$.
- A function that takes as input training input vectors and output values, and return the optimal weight vector $\hat{w}$.
- A function that takes as input a training set of input vectors and output values, and a test set input vectors, and output values, and returns the mean training error (i.e. average squared-error over all training samples) and mean test error.

b. To test your linear regression model, use part of the data set as a training set, and the rest as a test set. For each training set size, use the first lines of the data file as a training set, and the remaining lines as a test set. Write a function that takes as input the complete data set, and the desired training set size, and returns the mean training and test errors. Report the mean squared training and test errors for each of the following training set sizes: 10, 50, 100, 200, 300, 400.

c. Do the training and test errors tend to increase or decrease as the training set size increases? Why? Try some other training set sizes to see that this is only a tendency, and sometimes the change is in the different direction.

Question 2: LDA (MDA)

For this problem, load the data in P3.mat.
If you use Python, you can load the file using the following commands:

```
from scipy.io import loadmat
db = loadmat('P3.mat')
```

<u>Data</u>**:** This data consists of 3 classes from real plant data. Each sample has 4 features corresponding to measurements on plants. Each class corresponds to a different type of a plant.

a. Write a function *[Y,V] = lda(class1,class2,class3,dim)* which takes as an input 3 matrices, each holding samples from a single class piled as rows. The input dim should be an integer equal to either 1 or 2, to project the samples either to 1 or 2 dimensions (remember that for 3 classes, we can only project the data to 1 or 2 dimensions). The function should perform LDA and output the reduced samples in Y and the projection matrix in V. You can use Matlab function *eig* or *eigs*, or Python function *numpy.linalg.eig*.

b. Use the function you wrote in part (a) to project the data to 1 and 2 dimensions. Visualize your results separately for each dimension using function *scatter* in Matlab or *matplotlib.pyplot.scatter* in Python and different color for each class. Are the samples well separated in one dimension? In two dimensions?

c. Assume in the low dimensional space features have Gaussian distribution. Use leave-one-out cross-validation to compute the confusion matrix when using LDA to project to dimension 1 and dimension 2.

d. Use MSE for multiple classes to classify the samples, using again leave-one-out cross validation. Compute the confusion matrix. Discuss the difference (if any) between (c) and (d).

Question 3: Boosting

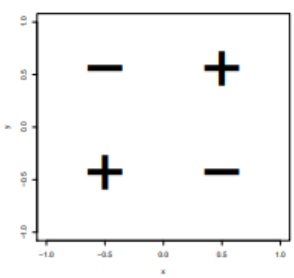

Figure 1: XOR dataset

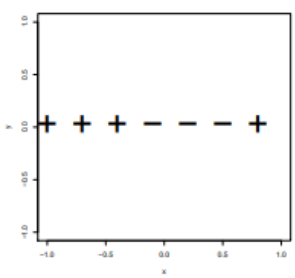

Figure 2: Seven point dataset

a. Suppose you have the 2-dimensional dataset depicted in Figure 1. Will Adaboost ever achieve better than 50% classification accuracy on this dataset? Why or why not? Briefly justify your answer.
b. Draw the decision boundary of the first classifier, h_1. Indicate which side is classified as the + class.
c. Compute and report ε_1 and α_1. What is the classification accuracy if we stop Adaboost here?
d. What are the new weights for the seven points?
e. Draw the decision boundary of the second classifier, h_2. Again, indicate which side is classified as the + class.
f. Which point(s) will have the lowest weight after the second iteration of Adaboost is finished?
g. Does the classification accuracy improve between first and second iterations of Adaboost? Explain briefly why the accuracy does (or does not) improve.

Question 4: Decision Trees

a. Consider the following dataset:

price	maintenance	capacity	airbag	profitable
low	low	2	no	yes
low	med	4	yes	no
low	low	4	no	yes
low	high	4	no	no
med	med	4	no	no
med	med	4	yes	yes
med	high	2	yes	no
med	high	5	no	yes
high	med	4	yes	yes
high	high	2	yes	no
high	high	5	yes	yes

Considering 'profitable' as the attribute we are trying to predict, which attribute would you select as the root in a decision tree with multi-way splits using the cross-entropy impurity measure? Explain.

b. For the same data set, suppose we decide to construct a decision tree using binary splits and the Gain impurity measure. Which among the following feature and split point combinations would be the best to use as the root node assuming that we consider each of the input features to be unordered? Explain
   a. price - {low, med}|{high}
   b. maintenance - {high}|{med, low}
   c. maintenance - {high, med}|{low}
   d. capacity - {2}|{4, 5}

c. Which of the following properties are characteristic of decision trees? Explain.
   a. High bias
   b. High variance
   c. Lack of smoothness of prediction surfaces
   d. Unbounded parameter set