Problem Set 6

MAS 622J/1.126J: Pattern Recognition and Analysis

Due Monday, 22 November 2010. Resubmission due, 24 November 2010

Note: All instructions to plot data or write a program should be carried out using MATLAB. In order to maintain a reasonable level of consistency and simplicity we ask that you do not use other software tools.

If you collaborated with other members of the class, please write their names at the end of the assignment. Moreover, you will need to write and sign the following statement: "In preparing my solutions, I did not look at any old homeworks, copy anybody's answers or let them copy mine."

Problem 1: Neural Nets [40 points]

In this problem, we are going to build a classifier to recognize handwritten digits. The datasets can be downloaded from the course webpage, they are from the UCI Machine Learning Repository:

http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

The 32 x 32 bitmaps of handwritten digits are preprocessed and are divided into non-overlapping blocks of 4 x 4 and the number of on-pixels are counted in each block. This generates an input matrix of 8 x 8 where each element is an integer in the range [0..16]. This reduces the dimensionality and gives invariance to small distortions.

The dataset was further processed, to rescale the input elements to a number between 0-1, and to transform the output values into 10 binary outputs. You get two arrays, data (input to neural network) and labels (output of neural network). The labels matrices contain the classification for the examples, as binary flags with zeros everywhere except for a 1 in the correct position (i.e. 0 0 0 0 0 0 1 0 0 0 is a 6 - positions correspond to: [0 1 2 3 4 5 6 7 8 9]).

For this problem you are free to write your own code or use any MATLAB toolboxes available for the purpose. You can use the default Neural Network toolbox available. Try help nnet, help nntool to help you get started. If you type demo on the MATLAB prompt, under the option Toolboxes, you can select Neural Networks to view some examples.

a. Train a two-layer neural network with sigmoidal hidden units (i.e. 1-hidden layer).

Train the network using the back-propagation algorithm with the provided training set. Test your network and report recognition results.

b. Experiment with different numbers of hidden units to optimize recognition accuracy.

c. Find a fixed number of hidden units, n that works well for the dataset and report the recognition results.

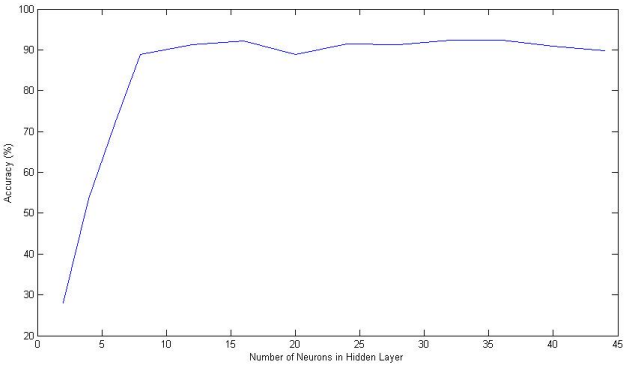
d. Comment on the effects of varying the number of hidden units on recognition accuracy. Suggestion: Plot the average percentage of recognition accuracy as a function of the number of neurons in the hidden layer to support your argument.

Solution:

```
load test
load train

trainingin = traindata';
trainingout = trainlabels';
testingin = testdata';
testingout = testlabels';
correct = zeros(20,1);
transfs = {'logsig'};
nums = [20 22 24 36 40 42 46 48 50 52];
for n=1:size(nums,2)
nh = nums(n);
layersizes = [nh];
mynet = newff(trainingin, trainingout, layersizes);
mynet.trainParam.epochs = 75;
mynet = train(mynet, trainingin, trainingout);
fprintf(1,'\nTraining done, n=%d; Testing...',nh);
%test
correct(nh) = 0;
for i=1:size(testingout,2)
testin = testingin(:,i);
out = sim(mynet,testin);
target = testingout(:,i);
    out;
    target;
x = argmax(out);
y = argmax(target);
if (x == y)
correct(nh) = correct(nh)+1;
end
end
fprintf(1,'\nperct correct =%d\n',((correct(nh)/size(testingin, 2))*100));
```

```
end
numhidden = argmax(correct);
```



Accuracy.jpg

Figure 1: Graph showing the accuracy of the 2 layer neural network with different numbers of neurons.

The graph shows that the accurcay is reaches a maximum at approximately 92%. Having greater than 12 neurons in the hidden layer does not show significant benefit.

e. Train a 2-hidden layer model using n/2 hidden units in each layer, where n corresponds to the value you found in part (c). Test your network and report recognition results. Compare and comment on the results of the 1-hidden layer model (part c) and the 2-hidden layer model (part e)

Solution: Well now train a network with two hidden layers, each layer having 6 units, by setting up the nnet in the following way:

```
transfs = {logsig logsig purelin};
layersizes = [6,6];
mynet = newff(trainingin, trainingout, layersizes);
mynet.trainParam.epochs = 100;
mynet = train(mynet, trainingin, trainingout);
twolayertest = 0;
for i=1:size(testingout,2)
testin = testingin(:,i);
out = sim(mynet,testin);
target = testingout(:,i);
x = argmax(out);
y = argmax(target);
if (x == y)
twolayertest = twolayertest+1;
end
end
fprintf(1,\nperct correct =%d\n,((twolayertest/size(testingin, 2))*100));
```

The performance of this network on the test data was 64.5%, less than the performance of a single hidden layer network with 12 neurons.

Problem 2: Decision Trees [40 points]

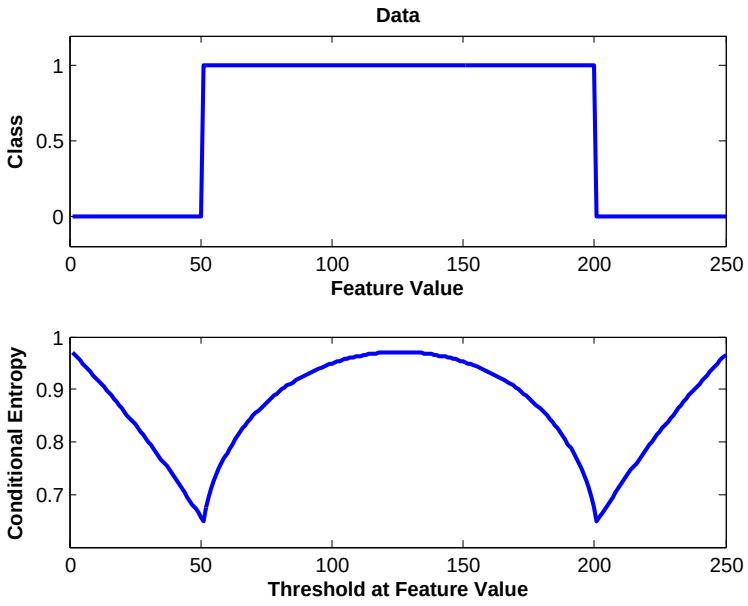
We collected the following information from people who were accepted into the doctoral program at MedianLab and people who did not.

Hardworking	Cool	GPA	Accepted
N	N	4	N
Y	N	3.6	Y
N	Y	3.4	Y
Y	N	3	N
Y	Y	3.1	Y
N	Y	2.9	Y

a. Assuming binary splits (e.g. GPA $< t$ vs. GPA $\geq t$), what values of t do you need to consider to find the optimal split of the feature GPA?

Solution:

As a general example, let's see what happens in a continuous feature with values from 1 to 250, where the values between 50 and 200 belong to class 1 and the rest of values belong to class 0:



Since we want to minimize the conditional entropy, there is no good by splitting the data if there are no transitions between classes. Therefore, the number of possible

Column i	1	2	3	4	5	6
Sorted GPA	2.9	3	3.1	3.4	3.6	4
Accepted	Y	N	Y	Y	Y	N

values of t is the number of transitions between two different classes in the sorted sequence of feature values.

In our case, there are three transitions between classes (column $i = 1$ to 2, 2 to 3, and 5 to 6). The values of t will be the middle values of GPA for each one of the transitions (i.e. 2.95, 3.05, and 3.8 respectively). Since we want to split each one of the transitions, we can assume that the middle values are the best boundaries.

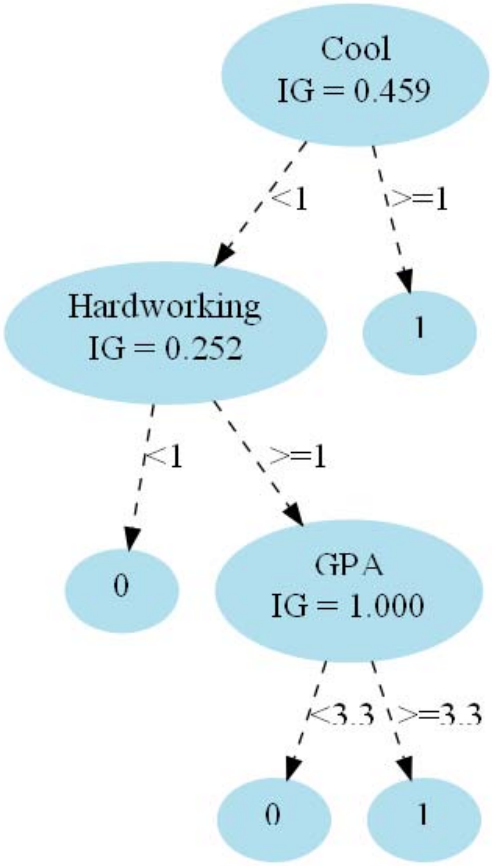
b. Write a program for training an ID3 decision tree. Draw the decision tree and indicate the information gain of each split. (Consider binary splits as in the previous question.)

Solution:

The MATLAB files can be downloaded from the course website. A brief summary of each function is as follows:

- entropy.m: This function computes entropy of a binary feature given one of the probabilities.
- computeEntropies.m: This function computes conditional entropies given a binary feature and their classes.
- obtainThresholds.m: This function finds the different threshold values for continues features following the criteria explained in the previous question.
- bestAttribute.m: This function computes the conditional entropies of an objective class given all the features in the dataset. The features can be either binary or continuous. Finally, the function finds the best split by computing the information gain of each split.
- buildTree.m: This is the recursive function that builds all the nodes in the ID3 algorithm. It also saves all the information in the file decision_tree.dot.
- ID3.m: It initializes the dataset, builds the decision tree by calling the function builTree.m, and generates the graph of the decision tree. For the latter step, we used the program Graphiz 2.26.3 (www.graphviz.org) which allows generating different types of diagrams.

The following figure shows the decision tree. IG means Information Gain, the leaf with value 1 is "Accepted = Y", and the leaf with value 0 is "Accepted = N".



Problem 3: Feature Selection [20 points]

Given the following observations of the class label Y:

X_1	X_2	X_3	X_4	Y
1	0	0	1	1
1	1	0	1	1
0	0	1	0	1
0	0	0	1	0
0	0	0	0	0
1	1	1	0	0

a. What is the mutual information of each feature with the class label?

Solution:

Mutual information is defined as:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log_2 \left(\frac{p(x,y)}{p_1(x)\, p_2(y)} \right)$$

The marginal probabilities that we need to compute are:

$$P(Y=0) = 3/6 = 0.5, P(Y=1) = 1 - P(Y=0) = 0.5$$
$$P(X_1=0) = 3/6 = 0.5, P(X_1=1) = 1 - P(X_1=0) = 0.5$$
$$P(X_2=0) = 4/6 = 0.667, P(X_2=1) = 1 - P(X_2=0) = 0.333$$
$$P(X_3=0) = 4/6 = 0.667, P(X_3=1) = 1 - P(X_3=0) = 0.333$$
$$P(X_4=0) = 3/6 = 0.5, P(X_4=1) = 1 - P(X_4=0) = 0.5$$

The joint probabilities that we need to compute are:

$$P(X_1=0,Y=0) = 2/6 = 0.333, P(X_1=0,Y=1) = P(X_1=0) - P(X_1=0,Y=0) = 0.167$$
$$P(X_1=1,Y=0) = 1/6 = 0.167, P(X_1=1,Y=1) = P(X_1=1) - P(X_1=1,Y=0) = 0.333$$
$$P(X_2=0,Y=0) = 2/6 = 0.333, P(X_2=0,Y=1) = P(X_2=0) - P(X_2=0,Y=0) = 0.333$$
$$P(X_2=1,Y=0) = 1/6 = 0.167, P(X_2=1,Y=1) = P(X_2=1) - P(X_2=1,Y=0) = 0.167$$
$$P(X_3=0,Y=0) = 2/6 = 0.333, P(X_3=0,Y=1) = P(X_3=0) - P(X_3=0,Y=0) = 0.333$$
$$P(X_3=1,Y=0) = 1/6 = 0.167, P(X_3=1,Y=1) = P(X_3=1) - P(X_3=1,Y=0) = 0.167$$
$$P(X_4=0,Y=0) = 2/6 = 0.333, P(X_4=0,Y=1) = P(X_4=0) - P(X_4=0,Y=0) = 0.167$$
$$P(X_4=1,Y=0) = 1/6 = 0.167, P(X_4=1,Y=1) = P(X_4=1) - P(X_4=1,Y=0) = 0.333$$

Using this information, the mutual information for each feature is: $MI(X_1,Y) = 0.0817$, $MI(X_2,Y) = 0$, $MI(X_3,Y) = 0$, $MI(X_4,Y) = 0.0817$.

b. What features would you choose to reduce the dimensionality? Is mutual information the best criteria for feature selection?

Solution:

In order to select a set of features, we can prioritize the ones with more mutual information because they are less independent to Y.

By looking at the results of the previous question, we can see that X_1 and X_4 are the features with more mutual information (0.0817), followed by X_2 and X_3 that do not have mutual information with Y (i.e. they are independent to Y). By inspection of the data, we can see that if we select X_1 and X_4 there are two samples of different classes with the same features ($X_1 = 0, X_4 = 0$). To avoid this problem, we can add X_3 as an extra feature.

Although the mutual information of X_3 with Y is zero, it does not mean that the combination of X_3 with other features will also have zero mutual information. Note that $Y = XOR(X_1, X_3)$.