Nguyen, Theodore

704-156-701

Math 156 Spring 2016

Lab 5: Ensemble learning

1(4) and 2 are attached as code .m files

3. (4) Use make cloud.m to generate data sampled from two point clouds, and use weaklearn.m to classify
the data directly (with uniform weights). How well did this classifier perform? Could a different single linear classifier perform better on this data? If so, what would the best case look like?

---- running the following code snippet

```
-----------------------------------------------------------------------------------------------------------------------------

[X0, X1] = make_cloud;

weakmodel = weaklearn(X0, X1);

X0weaktest = weakeval(X0, weakmodel);

X1weaktest = weakeval(X1, weakmodel);

num_incorrect = 0;

for i = 1:500

        if X0weaktest(i) == -1

                num_incorrect = num_incorrect + 1

        end

        if X1weaktest(i) == 1

                num_incorrect = num_incorrect + 1

        end

end

num_incorrect = 326             % out of 1000 total things to classify

--------------------------------------------------------------------------------------------------------------------------
```

showed that the weak classifier had a 32.6% misclassification rate, which is actually not that bad for a weak classifier since it was supposed to perform, at minimum, a 50% accuracy. I don't think another standalone linear classifier can perform much better on the data

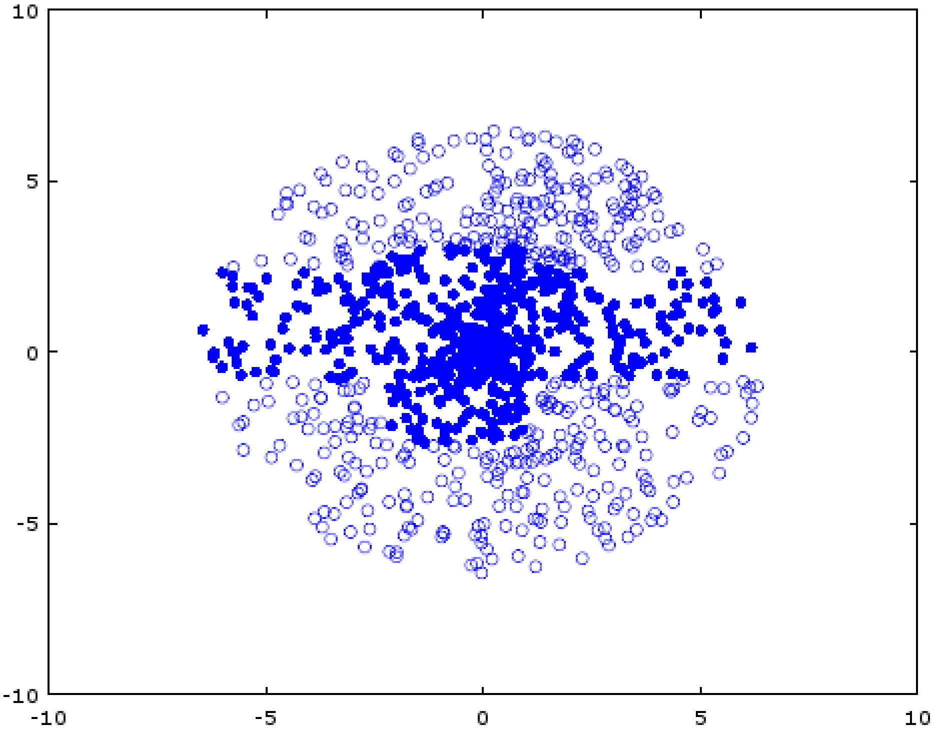
4. (4) Use boostlearn.m to classify the same data as above with M = 5 weak classifiers. Generate and turn in a figure showing the classification result (for example, use scatter).

---- Using essentially the exact same as the above script

```
[params, weights] = boostlearn(X0, X1, 5);

X0boost = boosteval(X0, params, weights);

X1boost = boosteval(X1, params, weights);

correctness(X0boost, X1boost)

ans = 270

----------------------------------------------------------------------------------------------------------------------
```

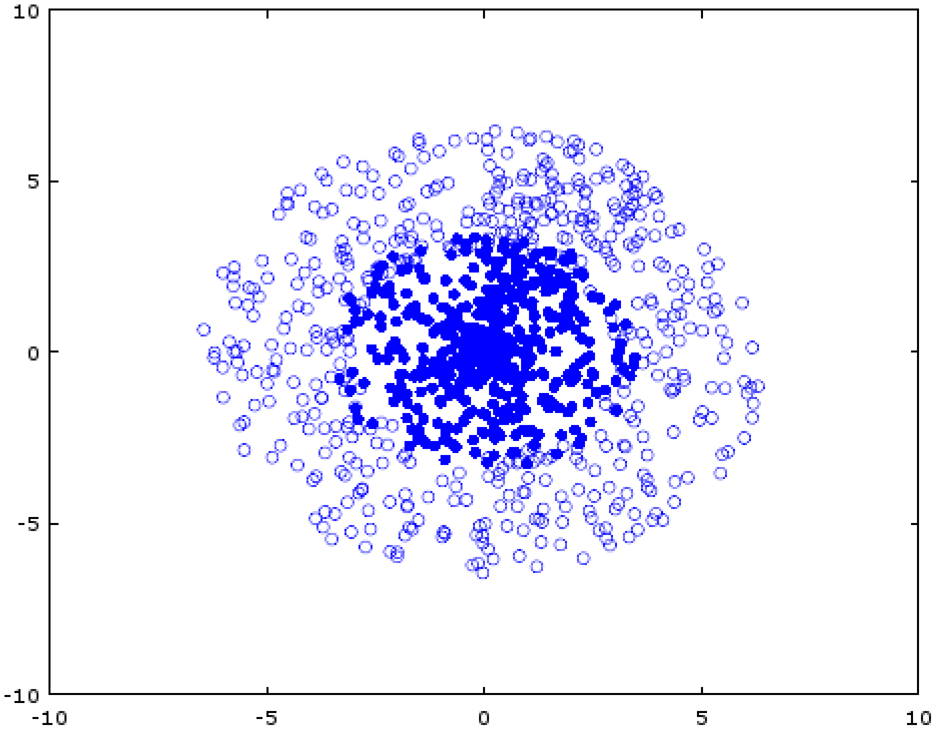
Where the function “correctness” is also included for convenience and counts the number of incorrectly classified observations. We have actually 270 observations classified incorrectly, which is about a 73% accuracy rate, only 10% higher than the weak classifier. This is surprising as we expected the aggregated model to be a much better classifier. This may be because we only used 5 weak classifiers in this case.

The following is a plot of the classification result:



Where the UNFILLED points are those that were classified as “-1” that is, into the X1 group, and the ones that are FILLED points were classified as “1” that is, into the X0 group.

BELOW is actually the TRUE classifications (basically just plotting X0 and X1), where the filled points are X0 points and the unfilled points are X1 points



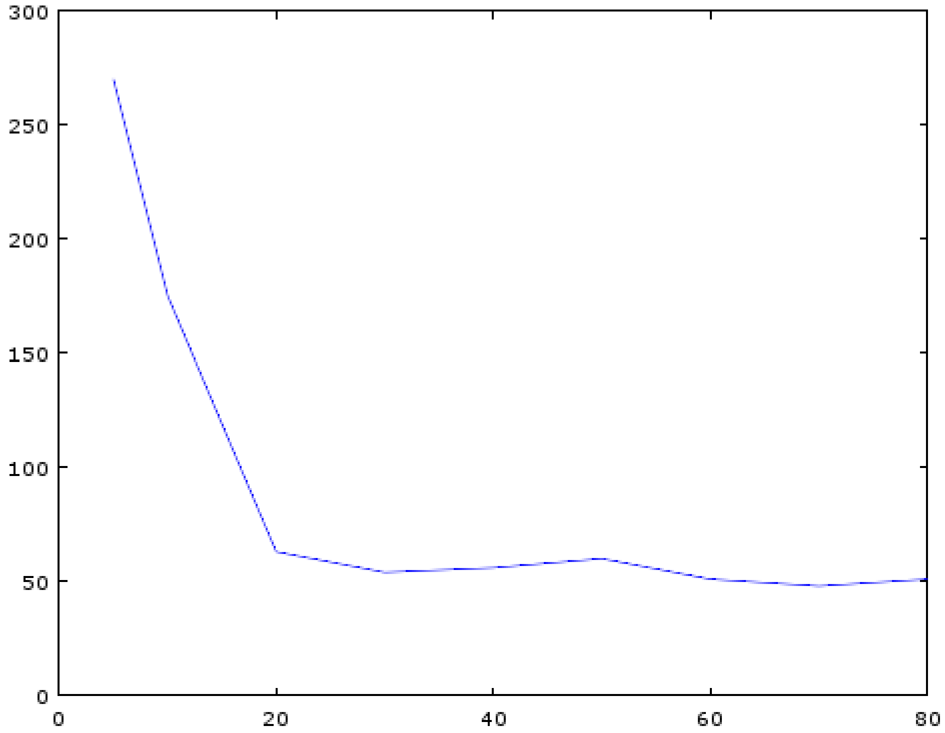
5. (4) Try larger values of M in the range 1 to 100 and study how the classifier changes. At what point do you believe the classifier is sufficiently reflecting the data? At what point do you believe the classifier might be over-fitting?

--- I planning on getting the data of accuracy for every single value of M from 1 to 100, but I was running Octave, which is substantially slower than MATLAB, so I just obtained the accuracies for some values of M = {5, 10, 20, 30, 40, 50, 60, 70, 80}. The respective number of incorrect classifications were Numwrong = {270, 175, 63, 54, 56, 60, 51, 48, 51}; there were a total of 1000 observations, so divide each number by 1000 to get the proportion that was classified incorrectly. We see that we get great returns in on accuracy when M increases from 5 to 20, but notice there is much fall off after M=20. I believe the classifier is sufficiently reflecting the data around and greater than M = 20. I think the model begins to overfit around M=40 or M=50, because the number the model classifies incorrectly increases around that timeframe.

6. (4) Use make cloud.m to generate more observations and test values for M. That is, generate and turn in a plot with M on the x-axis and the mis-classi_cation rate for new data on the y-axis. Describe the shape of this graph.

--- I just plotted the above data we talked about in question 5. Preferably, I would have generated and analyzed more data, but Octave is very slow, so I would assume that what I have is sufficient.

On the X-axis is the value of M, and on the Y-axis is the number of observations classified incorrectly for the model of that value of M. Note that this is the COUNT not the PROPORTION of observations classified incorrectly. A total of 1000 observations were classified for each value of M.



300
250
200
150
100
50
0
0
20
40
60
80