Stephen Chambers
CS830

Assignment 10 Writeup

1. **Describe any implementation choices you made that you felt were important. Mention anything else that we should know when evaluating your program. If you implement anything beyond the assignment as written, please be sure to discuss it**
   For Naïve Bayes, I kept a three dimensional array to keep track of the counts as I was reading in input. This drastically sped up my computations of P(xi|H). Overall, this sped up my program approximately 10x.

2. **What can you say about the the time and space complexity of your program?**

   KNN:
   Time complexity: O(dN)
   Space complexity: O(dN)

   Linear:
   Time complexity: $O(d^2N)$
   Space complexity: O(d)

   Naïve Bayes:
   Time Complexity: O(d*c)
   Space Complexity: O(c*d*v)

   Where:
   d = number of dimensions
   N = number of data points
   c = number of classes
   v = number of values per dimension

3. **For each algorithm you implemented, describe a learning problem that you think it will perform poorly on and explain why.**

   KNN:
   Problems with a lot of noise would require you to crank up k. However, the more k's there are, the longer the algorithm would take and the less accurate it would be. Therefore, knn would perform poorly on a large problem with a lot of noise.

   Linear:
   Linear would perform poorly on problems with inherent non-linearity. Additionally, problems with high dimensionaility would cause lienar regression to perform poorly, and it is easy to see why when looking at its time complexity.

Naïve Bayes:
Naïve Bayes assumes that the dimensions and their probabilities are independent of each other. If the probabilities were *not* independent, but rather very *dependent* on each other, Naïve Bayes would perform poorly.

4. **What suggestions do you have for improving this assignment in the future?**

   Please specify that the class_tool requires the agent to immediately write the classification to stdout after reading in each individual test data. It hangs if you read in all the test data first and then try to classify them one by one. Almost everyone I talked to had this problem. The assignment itself was really cool, I love this kind of stuff.

I included a short run through of the program running against the class_tool below, along with how long each algorithm took.

KNN

Summary:

Trial 1 -- size 10: accuracy 23.4%
Trial 2 -- size 50: accuracy 57.9%
Trial 3 -- size 100: accuracy 62.6%
Trial 4 -- size 500: accuracy 84.0%
Trial 5 -- size 1000: accuracy 88.8%
Trial 6 -- size 5000: accuracy 92.8%
Trial 7 -- size 10000: accuracy 94.8%
Trial 8 -- size 20000: accuracy 96.4%
Trial 9 -- size 58000: accuracy 97.0%

real 2m20.346s
user 2m37.360s
sys 0m3.292s

Linear Regression

Summary:

Trial 1 -- size 10: accuracy 28.1%
Trial 2 -- size 50: accuracy 46.4%
Trial 3 -- size 100: accuracy 59.0%
Trial 4 -- size 500: accuracy 73.7%
Trial 5 -- size 1000: accuracy 76.9%
Trial 6 -- size 5000: accuracy 79.8%
Trial 7 -- size 10000: accuracy 82.3%
Trial 8 -- size 20000: accuracy 83.1%
Trial 9 -- size 58000: accuracy 81.4%

real 0m23.675s
user 0m40.355s
sys 0m2.593s

Naïve Bayes

Summary:

Trial 1 -- size 10: accuracy 28.3%
Trial 2 -- size 50: accuracy 54.1%
Trial 3 -- size 100: accuracy 61.0%
Trial 4 -- size 500: accuracy 79.2%
Trial 5 -- size 1000: accuracy 80.6%
Trial 6 -- size 5000: accuracy 81.5%
Trial 7 -- size 10000: accuracy 83.8%
Trial 8 -- size 20000: accuracy 84.9%
Trial 9 -- size 58000: accuracy 85.0%

real 0m23.107s
user 0m42.212s
sys 0m3.062s