Introduction to Machine Learning *Ananth, AP68098*

HW1: Decision trees and KNN

Please note that only PDF submissions are accepted. We encourage using LaTeX to produce your writeups. You'll need *mydefs.sty* and *notes.sty* which can be downloaded from the course page.

1. (not graded): The following are true/false questions. You don't need to answer the questions. Just tell us which ones you can't answer confidently in less than one minute. (You won't be graded on this.) If you can't answer at least 8, you should probably spend some extra time outside of class beefing up on elementary math. I would strongly suggest going through this math tutorial by Hal Daume: http://www.umiacs.umd.edu/~hal/courses/2013S_ML/math4ml.pdf

   (a) $\log x + \log y = \log(xy)$

   (b) $\log[ab^c] = \log a + (\log b)(\log c)$

   (c) $\frac{\partial}{\partial x}\sigma(x) = \sigma(x) \times (1 - \sigma(x))$ where $\sigma(x) = 1/(1 + e^{-x})$

   (d) The distance between the point (x_1, y_1) and line $ax + by + c$ is $(ax_1 + by_1 + c)/\sqrt{a^2 + b^2}$

   (e) $\frac{\partial}{\partial x} \log x = -\frac{1}{x}$

   (f) $p(a \mid b) = p(a, b)/p(b)$

   (g) $p(x \mid y, z) = p(x \mid y)p(x \mid z)$

   (h) $C(n, k) = C(n-1, k-1) + C(n-1, k)$, where $C(n, k)$ is the number of ways of choosing k objects from n

   (i) $||\alpha \boldsymbol{u} + \boldsymbol{v}||^2 = \alpha^2 ||\boldsymbol{u}||^2 + ||\boldsymbol{v}||^2$, where $||\cdot||$ denotes Euclidean norm, α is a scalar and $\boldsymbol{u}$ and $\boldsymbol{v}$ are vectors

   (j) $\left|\boldsymbol{u}^\top \boldsymbol{v}\right| \geq ||\boldsymbol{u}|| \times ||\boldsymbol{v}||$, where $|\cdot|$ denotes absolute value and $\boldsymbol{u}^\top \boldsymbol{v}$ is the dot product of $\boldsymbol{u}$ and $\boldsymbol{v}$

   (k) $\int_{-\infty}^{\infty} \mathrm{d}x \exp[-(\pi/2)x^2] = \sqrt{2}$

2. (not graded): Go though this matlab tutorial by Stefan Roth:
   http://cs.brown.edu/courses/csci1950-g/docs/matlab/matlabtutorialcode.html

3. How many decision trees are there with 3 binary attributes? With 4?

   > Ans: For Three: 256, for Four: 65536. 2^{2^n} number of distinct Decision Trees are possible, where n is the number of binary attributes.

4. In class, we looked at an example where all the attributes were binary (i.e., yes/no valued). Consider an example where instead of the attribute "Morning?", we had an attribute "Time" which specifies when the class begins.

   (a) We can pick a threshold τ and use (Time $< \tau$)? as a criteria to split the data in two. Explain how you might pick the optimal value of τ.

   > Function is to sort the training data by feature value and then test split values that are the mean of ordered training points.

   (b) In the decision tree learning algorithm discussed in class, once a binary attribute is used, the sub-trees do not need to consider it. Explain why when there are continuous attributes this may not be the case.

After splitting the tree based on (say) temperature, we might still need to split further for classification. Suppose, we consider the upper half (assume 0-30) of temperature after splitting, we might have to again split for $temperature < 10$. Here the temperature is again used down further in the tree for classification.

5. Why memorizing the training data and doing table lookups is a bad strategy for learning? How do we prevent that in decision trees?

The program is most likely to predict correctly every time because it has already seen the data, i.e. Memorized the data. This can be prevented by splitting the data into *training data* and *testing data*. Let the program learn through training data, be tested using testing data (which it has not seen before).

6. What does the decision boundary of 1-nearest neighbour classifier for 2 points (one positive, one negative) look like?

The decision boundary will be the mid-point of the distance between the two points.

7. Does the accuracy of a kNN classifier using the Euclidean distance change if you (a) translate the data (b) scale the data (i.e., multiply all the points by a constant), or (c) rotate the data? Explain. Answer the same for a kNN classifier using Manhattan distance[1].

A) Translate the data: The accuracy will remain constant when all the points with respect to each other is moved (Addition) by a constant amount.
B) Scale the data: The accuracy will remain constant when scaled (Multiplied) by a constant. Accuracy depends only on the distance between two points.
C) Rotate the data: The accuracy will remain constant when the direction of all the points with respect to each other is reversed. (transpose).
The accuracy varies for Manhattan Distance when data is scaled, translated or rotated. The absolute difference between the coordinates changes.

8. Implement kNN in matlab for handwritten digit classification and submit all codes and plots:

   (a) Download MNIST digit dataset (60,000 training and 10,000 testing data points) and the starter code from the course page. Each row in the matrix represents a handwritten digit image. The starter code shows how to visualize an example data point in matlab. The task is to predict the class (0 to 9) for a given test image, so it is a 10-way classification problem.

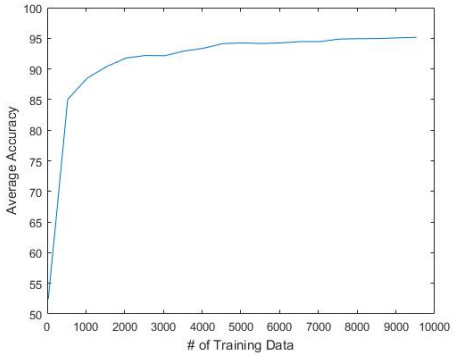
   (b) Write a matlab function that implements kNN for this task and reports the accuracy for each class (10 numbers) as well as the average accuracy (one number).
   [acc acc_av] = kNN(images_train, labels_train, images_test, labels_test, k)
   where *acc* is a vector of length 10 and *acc_av* is a scalar. Look at a few correct and wrong predictions to see if it makes sense.

script.m, kNN.m

[1] `http://en.wikipedia.org/wiki/Taxicab_geometry`

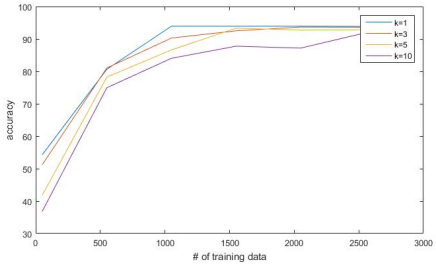
(c) For $k = 1$, change the number of training data points (30 to 10,000) to see the change in performance. Plot the average accuracy for 10 different dataset sizes. You may use command *logspace* in matlab. In the plot, x-axis is for the number of training data and y-axis is for the accuracy.

EigthC.m



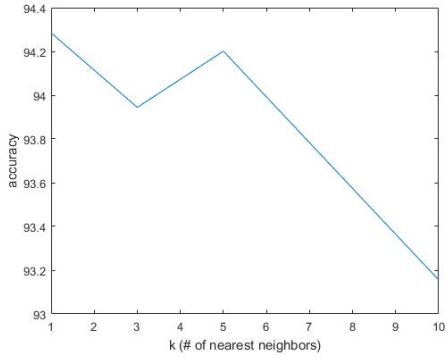
(d) Show the effect of k on the accuracy. Make a plot similar to the above one with multiple colored curves on the top of each other (each for a particular k from 1 to 10.) You may use command *legend* in matlab to name different colors.

EigthD.m



(e) Choose the best k for 10,000 total training data by splitting the training data into two halves (the first for training and the second for validation). You may plot the average accuracy wrt k for this. Note that in this part, you should not use the test data.

EigthE.m


94.4
94.2
94
93.8
93.6
93.4
93.2
93
accuracy
1
2
3
4
5
6
7
8
9
10
k (# of nearest neighbors)