Introduction to Machine Learning *Mohit Khatwani, AJ75499*

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

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

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

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

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

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

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

   (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 TRUE

   (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 $\boldsymbol{v}$ FALSE : I think sign should be reversed

   (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. 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.

   The optimal value of τ will be when the dataset is split in two halves and gives most number of correct answers. One way of calculating τ can be as follows:

   - In order to perform calculation with respect to time we should convert time in proper scalar format. One way can be to represent time in minutes, starting from 00:00 i.e. 12 am.
   - First list down all the values for which student liked the subject or he gave a positive rating.
   - After listing these values calculate maximum value of time for which he liked the subject
   - Secondly, list down all the values of time for which student didn't liked the subject and calculate minimum value
   - After getting two values of time from lists. Calculate average of those values. This average value can be represented as τ

Various measures are also available in order to find whether τ is a best split which maximum information gain Ex. Gini-Index, Entropy.
The idea behind selecting optimal τ is by selecting a bucket for which our classifier gives maximum number of correct answers for that feature.

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

For binary attribute the possible outcomes are judged in terms of yes or no. If one output is selected then there is no possibility of getting the resulting label from non-selected value. That is not the case with continuous attributed values. In continuous attributed feature we select a splitting criteria and form buckets for different ranges of values.
According to these buckets we find out for which condition we get optimum result.
Thus for continuous attributes the sub-tree needs to consider the feature.
In real world problems there is a probability assigned to every criteria in attribute.It may happen for present node there is a higher probability of selecting a specific bucket, but it may change as more attributes are taken into consideration.
One example can be used as route finding problem. In this problem more than two options are available for every node in decision tree, thus sub-tree needs to consider every feature in order to find an optimal solution.

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

- There are various disadvantages of memorizing training data. Training data is very large in size, therefore it is very inefficient to store that much data
- An algorithm which memorizes the training data will only work if it comes across a test data which is already seen or available in training data by performing lookups
- In order for our algorithm to work for most of the test data, we need to generalize our algorithm by minimizing our loss function
- In decision tree classifier over-fitting can be prevented by restricting levels or height of our decision tree
- Height of decision tree acts as a hyper-parameter which is used to validate the algorithm

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

For 1-nearest neighbour let us consider two points which are plotted along Y-axes One of the points is negative and other is positive. So for finding a decision boundary we must find the set of points from which distance of 2 points given is same.
Another way can be to find the slope of the line by joining these two points and consider a perpendicular bisector to this line which will act as our decision boundary.

6. 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 the all the points by a constant), or (c) rotate the data? Explain. Answer the same for a kNN classifier using Manhattan distance[1].

(a) For Euclidean Distance: Euclidean distance is represented as $\sqrt{\Sigma(x1-x2)^2...}$ for all dimensions

i. Transformation of Data
Transformation of data means applying some function on data vector.
Suppose data vector is in form (x,y), then after data transformation (*f(x),f(y)*)

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

Euclidean distance is used as distance metrics for comparison and finding nearest neighbors. By transforming data, euclidean distance will change but it will not effect the nearest neighbors as relative distance between them will be same.
Accuracy will not be affected in Transformation of data.

   ii. Scaling the Data
   Scaling the data means the vector will be multiplied by a constant without change in direction.
   Assuming scaling factor is a.
   change in euclidean distance will be as $\sqrt{\Sigma(ax1 - ax2)^2...} =$ a.$\sqrt{\Sigma(x1 - x2)^2...}$
   Here every euclidean distance will change with a same factor which will not affect the nearest neighbor.
   Accuracy will not change for Scaling of data with euclidean distance as distance metric. Here assuming that scaling factor is not ZERO.

   iii. Rotating the Data
   The data is rotated about the origin. Here there will be no difference in euclidean distance.
   As there is no change in euclidean distance, rotating the data will not affect the accuracy of KNN algorithm.

(b) For Manhattan Distance:
Manhattan Distance is defined as sum of absolute values of X-intercept and Y-intercept and so-on...
$\Sigma \mid x1 - x2 \mid + \mid y1 - y2 \mid ...$

   i. Transformation of Data
   Here even if some transformation function is applied on data. The relative Manhattan distance will remain constant.
   Suppose we use a function (x) which transforms/shifts data along X-axis.
   Here the relative absolute values of X-intercept will not change.
   There will be no effect in accuracy in transformation of data while using manhattan distance.

   ii. Scaling the Data
   Suppose we apply a sclaling factor of 'a' on our data.
   After scaling factor manhattan distance will change as follows. a $\mid x1 - x2 \mid$ Here there will be no change in relative manhattan distance, thus accuracy will not be affected.

   iii. Rotating the Data
   In order to explain this lets consider an example from taxicab geometry.
   A circle is represented as points which are equidistant from centre.
   In taxicab geometry this representation changes as we consider manhattan distance here.
   A circle can denoted by polar co-ordinates as R $\mid cos\theta + sin\theta \mid$
   Here manhattan distance depends on the angle in which it is considered , thus rotation will effect the accuracy of KNN algorithm.

7. Implement kNN in Matlab or Python 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 or Python 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. To speed it up, in all experiments, you may use only the first 1000 testing images.

(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.

(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 in [1 2 3 5 10].) You may use command *legend* in matlab to name different colors.

(e) Choose the best k for 2,000 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. You may search for k in this list: [1 2 3 5 10].

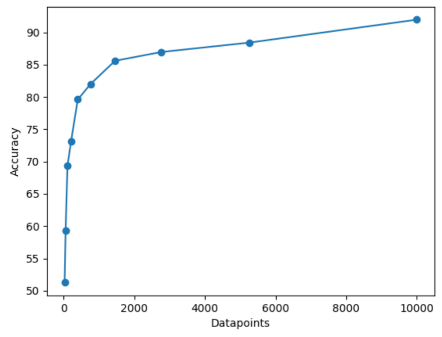

Figure 1: This graph denotes increase in Accuracy along different sizes of training datapoints ranging from 30 to 10,000

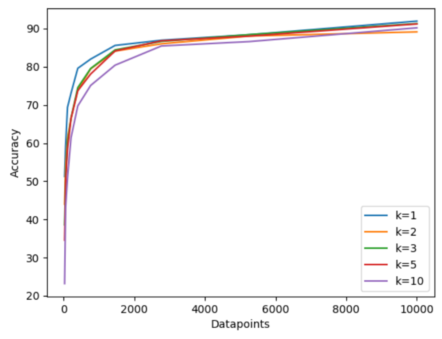

Figure 2: This graph denotes accuracy along different training data sizes with different values of K ranging in [1,2,3,5,10]

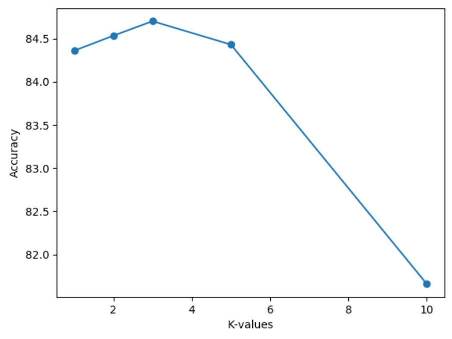

Figure 3: This graph denotes accuracy for different values of K