Machine Learning & Discovery *ECS 271, Fall 2022*

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. You can use overleaf.com to write your latex file and save the result as PDF.

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: https://web.cs.ucdavis.edu/%7Ehpirsiav/courses/MLf22/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) $\int_{-\infty}^{\infty} \mathrm{d}x \exp[-(\pi/2)x^2] = \sqrt{2}$

I could answer the questions confidently within a minute.

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.

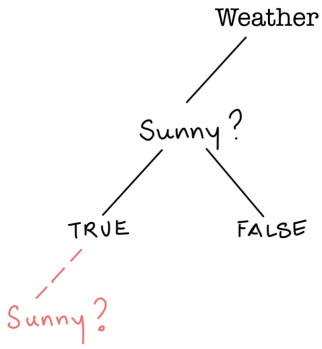
   (a) We can pick a threshold τ and use (Time $< \tau$)? as a criteria to split the data in two. Explain how you might pick the optimal value of τ.

In class, we looked at an example with a discrete variable (Morning?). Unlike the Morning variable, the example provided in this question, "Time," has continuous values. Selecting a τ threshold for the variable is crucial to divide the dataset into two parts. The steps listed below can be used to choose the optimal value of tau:

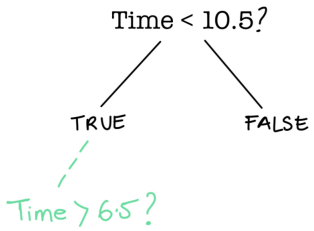
- Sort the set of values that the variable can take (observed data) in ascending order
- Based on intuition, we now know that the classification error at any point between two values will be the same
- With this intuition in mind, we use the mid-point of each pair of values in our sorted list and calculate the information gain/entropy using this mid-point value as the threshold
- The ideal τ to divide the dataset by will be the value that provides the most information gain or the least entropy.

(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.

Once an attribute has been evaluated for binary variables, the subtree does not need to be considered again. To illustrate this, Say the decision tree used the Sunny? attribute and got a true response. It would not make sense to consider Sunny? again because there would be no information gained since we already know that Sunny? has been evaluated to true.



However, the situation is different in the case of continuous attributes. Reusing the same attribute within the subtree has advantages because of the wide variety of values that an attribute can accept. For instance, the time attribute can be queried again for a new range of values to narrow down the class if it was initially split based on the value of 10.5 seconds in the time attribute.

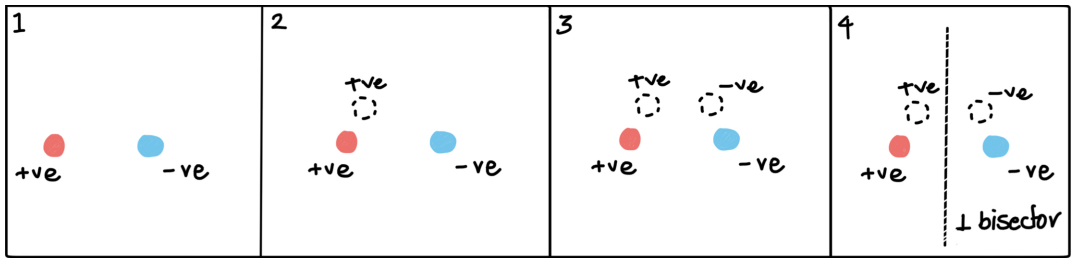


4. Why memorizing the training data and doing table lookups is a bad strategy for learning? How do we prevent that in decision trees?

> The end goal of a machine learning algorithm is to generalize the insights/learnings inferred from the training data and apply them to unseen test data. The fundamental goal of machine learning is lost by memorizing the training data and using lookup tables. Using such lookup tables leads to overfitting of the data, in which there is a direct mapping between the seen data and the output, which is antithetical to what machine learning is trying to achieve. Additionally, overfitting learns specifics about outliers or noisy data, which makes it more complex to generalize.
> Overfitting in decision trees happens when we partition the dataset until all possible outcomes are seen. Although the decision tree's accuracy in this situation would be almost perfect, it would not be able to generalize its results when testing with unseen data. We can employ strategies to either stop the dataset from being split or prune branches once the tree has been built to avoid this. The train data set is divided into train data and validation data to prevent overfitting. We can stop the tree from repeatedly splitting the data until it becomes overfit by regulating the depth of the tree. The decision tree's hyperparameter, or the depth of the tree, can be adjusted using the validation data to ensure the model is not overfit.

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

> Let's say there are two points, one with a positive label (in red) and one with a negative label (in blue). We view this as a 1-nearest neighbor classifier, or $k = 1$, as mentioned in the question. A new data point that is nearer to the positive will be labeled as positive (see step 2 in the diagram). Similar to this, a new data point that is closer to the negative (see step 3) will be labeled as negative. The decision boundary is formed by a perpendicular bisector (see step 4) of the hypothetical line connecting the two initial points. As a result, the classification of a new point is determined by which side of the perpendicular bisector it lies on.



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].

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

(a) Given the mathematical representation for translating the data to be:

$$f(x, y) = [(x_1 + c_1), (y_1 + c_2)]..., [(x_n + c_1), (y_n + c_2)]$$

where f is the translation function, $x_1, y_1, x_2, y_2..., x_n, y_n$ are the existing data points and c_1, c_2 are the constants that are used to translate the data points. Since all the points are moving by c_1, c_2, the relative euclidean distance between points remain the same. As a result, the kNN classifier's accuracy will not change.

(b) Given the mathematical representation for scaling the data to be:

$$f(x, y) = [(cx_1, cy_1)]..., [(cx_n, xy_n)]$$

where f is the scaling function, $x_1, y_1, x_2, y_2..., x_n, y_n$ are the existing data points and c is the constant used to scale the data points. Since all the points are scaled by a factor of c, the resulting distance of the points would also be affected proportionally. Since the euclidean distance is relatively the same – the accuracy of the kNN classifier won't change.

(c) Given the mathematical representation for rotating the data to be:

$$f(x, y) = R(\theta) \times \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

where f is the rotation function, $R(\theta)$ is the rotation matrix and x_n, y_n are the existing data points. On rotation of the data points, the euclidean distance between the points do not change. Since there is no change in distance the accuracy of the kNN classifier won't change.

A kNN classifier using Manhattan distance would follow the same conditions as above .i.e, regardless of whether the data points are translated, scaled or rotated – the accuracy of the classifier will remain the same.

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. First choose 2,000 training data randomly (to speed up the experiment). Then, split the training data randomly to two halves (the first for training and the second for cross-validation to choose the best k). Please plot the average accuracy wrt k on the validation set. You may search for k in this list: [1 2 3 5 10]. Finally, report the accuracy for the best k on the testing data.