CS 688: Machine Learning
George Mason University, Fall 22 **Homework 4** **Due at 11:59pm, Nov 29**

Problems marked with **E** are graded on effort, which means that they are graded subjectively on the perceived effort you put into them, rather than on correctness. We strongly encourage you to typeset your solutions in LaTeX.

1. (E 40 points) In this problem you will apply a K-means algorithm to compress image of size $N \times N$ into blocks of size $M \times M$.
   (a) Download `mandrill.png` and `hw4_p1`. Given $M = 2$, $K = 64$, Partition the image into blocks of size $M \times M$ and reshape the block into a vector of length $3 \cdot M^2$. (The 3 comes from RGB values for each pixel) Next, write K-means algorithm yourself with K that sample random data points (without replacement) as the initial cluster means. Finally, reconstruct a compressed version of the original image by replacing each block in the original image with the nearest center.
      i. Plot of the distortion measure
      $$J(\mu_1, \ldots, \mu_K; z_1, \ldots, z_n) = \sum_{j=1}^{n} \sum_{k=1}^{K} \mathbb{I}(z_j = k) \|x_j - \mu_k\|_2^2$$
      as the iteration increases where μ_k is the center of cluster $k \in [K]$ and z_j is block j's cluster assignment.
      ii. Show the compressed image.
   (b) The original uncompressed image uses 24 bits per pixel (bpp), 8 bits for each color. Given an image of size $N \times N$ with block size $M \times M$ and K clusters, what is the compression ratio, the ratio of bpp in the compressed image relative to the original uncompressed image? *Hint: To calculate the bpp of the compressed image, imagine you need to transmit the compressed image to a friend who knows the compression strategy as well as the values of M, N, and K*

2. (E 60 points) You will implement bagging and random forest on a handwritten digit dataset that has $N = 720$ examples from 4 classes. Specifically, implement functions `bagging_ensemble`, `random_forest`, and `majority_vote` in `hw4_p2`, and compare the performance of these ensemble classifiers using 100 random splits of the digits dataset into training and test sets, where the test set contains roughly 20% of the data. Run both algorithms on these data and obtain 100 accuracy values for each algorithm.
   (a) Given a dataset D of size n, a bootstrapped dataset contains n samples drawn randomly from D with replacement. What is the expected number of distinct elements normalized by n in a boostrapped dataset? Given two independent boostrapped datasets $\tilde{D}_1, \tilde{D}_2$ of D, what is the expected number of common element normalized by n that are in both $\tilde{D}_1$ and $\tilde{D}_2$? What are the limits as n goes to infinity?
   (b) A bagging ensemble classifier consists of `n_clf` decision trees where each decision tree is trained independently on a bootstrap sample of the training data, and the final prediction of the bagging classifier is a majority vote of these `n_clf` decision trees. (Note that `RandomForestClassifier` in sklearn uses means as the final prediction)
   (c) Implement random forest. Like bagging, random forest also consists of `n_clf` decision trees where each decision tree is trained independently on a bootstrap sample of the training data. However, for each node we randomly select m features as candidates for splitting on (see parameter `max_features` of `sklearn.tree.DecisionTressClassifier`). Again, here the final output is determined by majority vote.