Assignment 4
Classifying Digits

DUE: Wednesday, March 10, 2021 at 11:59 pm PST
Late assignments will **NOT** be accepted

Professor: Jason J. Bramburger

Download the MNIST data set (both training and test sets and labels): http://yann.lecun.com/exdb/mnist/
To load the MNIST data into MATLAB you can use the attached mnist_parse.m function and execute as follows:

```
[images, labels] = mnist_parse('train-images-idx3-ubyte', 'train-labels-idx1-ubyte');
```

The above command loads in the training data. You can use the same command for the test data by changing the file names on the right-hand-side.

Your job is to perform an analysis of this data set. You will start by performing the following analysis:

1. Do an SVD analysis of the digit images. You will need to reshape each image into a column vector and each column of your data matrix is a different image.
2. What does the singular value spectrum look like and how many modes are necessary for good image reconstruction? (i.e. what is the rank r of the digit space?)
3. What is the interpretation of the $\mathbf{U}$, $\mathbf{\Sigma}$, and $\mathbf{V}$ matrices?
4. On a 3D plot, project onto three selected $\mathbf{V}$-modes (columns) colored by their digit label. For example, columns 2,3, and 5.

Once you have performed the above and have your data projected into PCA space, you will build a classifier to identify individual digits in the training set.

- Pick two digits. See if you can build a linear classifier (LDA) that can reasonable identify them.
- Pick three digits. Try to build a linear classifier to identify these three now.
- Which two digits in the data set appear to be the most difficult to separate? Quantify the accuracy of the separation with LDA on the test data.
- Which two digits in the data set are most easy to separate? Quantify the accuracy of the separation with LDA on the test data.
- SVM (support vector machines) and decision tree classifiers were the state-of-the-art until about 2014. How well do these separate between all ten digits? (see code below to get started).
- Compare the performance between LDA, SVM and decision trees on the hardest and easiest pair of digits to separate (from above).

Make sure to discuss the performance of your classifier on both the training and test sets.

The following code is meant to illustrate how to implement SVM and decision tree classifiers in MATLAB using their built-in functions. You will need to change the variable names and potentially look into the documentation for each function. The decision tree function fitctree() documentation is found at this link and the SVM function fitcsvm() documentation is found at this link.

```
% classification tree on fisheriris data
load fisheriris;
tree=fitctree(meas,species,'MaxNumSplits',3,'CrossVal','on');
view(tree.Trained{1},'Mode','graph');
classError = kfoldLoss(tree)

% SVM classifier with training data, labels and test set
Mdl = fitcsvm(xtrain,label);
test_labels = predict(Mdl,test);
```