Michael Ing

ECE 2195 Hwk 2

Due 10/31/2018

Face Recognition SVM Classifier

My SVM Classifier is composed of two files, SVM\_Runner\_Linear.py and SVM\_Runner\_Poly.py, that reads the faces, trains, and tests sequentially. The first file has linear SVM implementations while the second file has polynomial kernel SVM implementations, and both utilize a one v. one algorithm with five-fold cross-validation.

Both files start by dividing the images into 5 random but equal subsets. Both then precede to choose a subset as test data, marking the remaining images as the training set. Using a one v. one classifier technique, both files start to train the 800 odd classifiers by running through the 8 training images per face, getting the pixel values into a flat list of size 10304 and using the cvxopt qp solver to get alpha values. For the linear file the kernel used is simply the x\*y\*x.T\*y.T, but the polynomial kernel file uses a more complex kernel based on the previously defined degree, taking y\*(x\*x.T + 1) ^ d, where d is the defined degree. These generated alpha values are used to calculate the weights and biases, and continued repetition over all of the images creates 800 or so one v. one classifiers. The testing data is now ready to be passed in.

Using the remaining 20% of the images, this testing data is passed through the proper equation (w.T\*x + b for the linear classifier, and K(x, w) + b for the polynomial kernel) and tested against 0 to see where it is classified with each classifier, either as class 1 if positive or class 2 if negative. This is repeated for each image for all 800 one v. one classifiers, and the results are kept as running totals in a list. Once each image has gone through every classifier, the image is categorized as the highest running total in the list, and this result is evaluated against the actual class. This procedure is repeated until all test images are finished testing and a total accuracy is calculated. Afterwards, the entire procedure is repeated for the other 4 subsets as test sets (to get the five-fold cross-validation), and a final average accuracy is reported.

After lots of testing, I was able to get a high accuracy on the linear SVM classifier of around 98.0% after five-fold cross-validation. Unfortunately, my accuracy for the polynomial kernel SVM classifier was much lower, hovering around 50% or 60%. I believe this was an issue not with training but with the classification process via the test data. While decent, I believe that further extensive testing would allow me to find the error in the test data kernel equation and raise the accuracy percentage.