ECE532 Final Project Update 2

Timothy Winfree

November 2020

1 Added Pre Processing steps

As a review, I am passing each image through 20 filters, the output of each being down sampled to 8x8 pixels. This dimension was determined to have the highest accuracy (90%) in cross validation, using a KNN classifier with k = 3 on a random set of 9000 training images and 1000 validation images. Thus, each image is described by a feature vector containing $8 * 8 * 20 = 1280$ elements. There is surely redundancy in using this many features. To trim redundant features, I first subtracted from each column its mean, then divided each column by the magnitude of its max value. Using the singular value decomposition, I projected X onto a 112 dimensional subspace defined by the 112 principal components of X with the highest singular values.

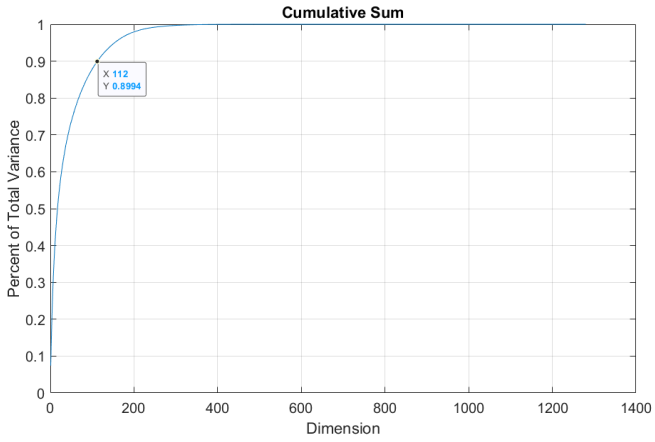


Figure 1: Cumulative sum of singular values normalized to sum of all singular values of X. Evidently, 90% of the variance in X can be captured by the first 112 principle components. A ten fold reduction in the size of X.

2 K-Nearest Neighbors

The K-nearest neighbors algorithm is simple to implement, and a natural multiclass classifier. To compute the label of a novel data point with feature vector $\hat{x}$, I compute the difference between $\hat{x}$ and each data point in the training data, then predict the label of $\hat{x}$ as the mode of the k nearest labels (euclidean distance). If there is a tie I apply a weight function $\frac{1}{d}$ to each vote to break the tie, where d is the distance of the voting data point to $\hat{x}$. I used cross validation to determine the best values for k and the dimensionality n. Using $k = 3$ and $n = 50$, I classified the test bank of 10,000 images with an accuracy of 96.8%.

3 One vs. One Binary Least Squares Classifiers

In order to use least squares for multiclass classification, I must split the problem up into a series of binary classification problems. To do this, I trained 45 one vs one binary classifiers, one for each way of choosing

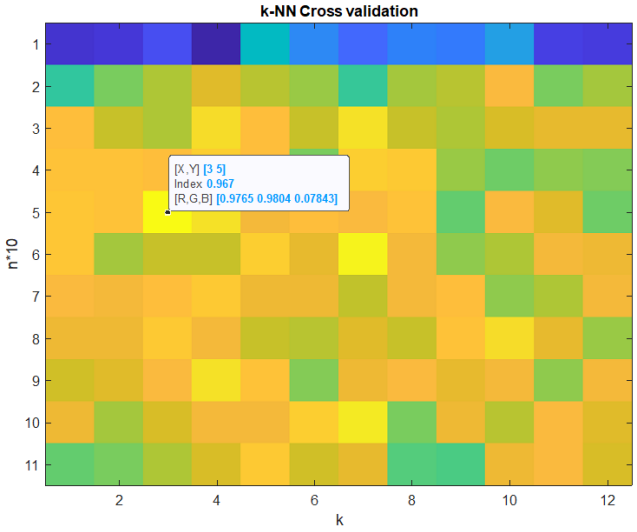


Figure 2: Randomly selecting 5 sets of 50,000 training and 10,000 validation images, I computed the average accuracy of the knn algorithm for the shown values of k and n.

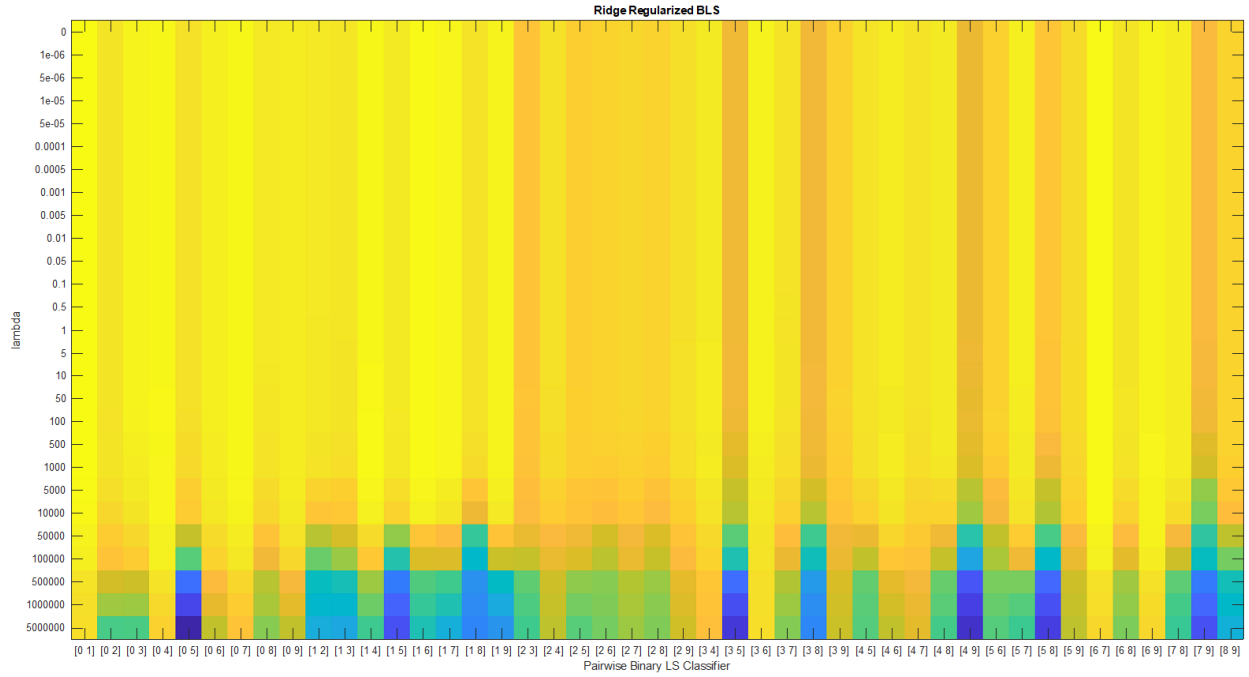
two classes from 10. Prediction of a novel data point with feature vector $\hat{x}$ takes place as follows. For each of the 45 classifiers, I compute $sign(\hat{x}^T \cdot w_n)$, the class assigned to this output (either 1 or -1) receives a vote. The class with the most votes is chosen as the prediction. If there is a tie, classifiers that involve only classes involved in the tie are used to decide the vote, e.g., if there is a tie between 4 and 9, the winner of the 4 vs. 9 classifier is chosen as the prediction. I performed cross validation separately for each of the 45 classifiers using both lasso and ridge. As shown in figure 3, none of the classifiers benefit from the presence of either regularizer. As such, I trained each classifier on the 60,000 training images with no regularizer and classified the 10,000 test images with an accuracy of 94.1%.

4 Next Steps

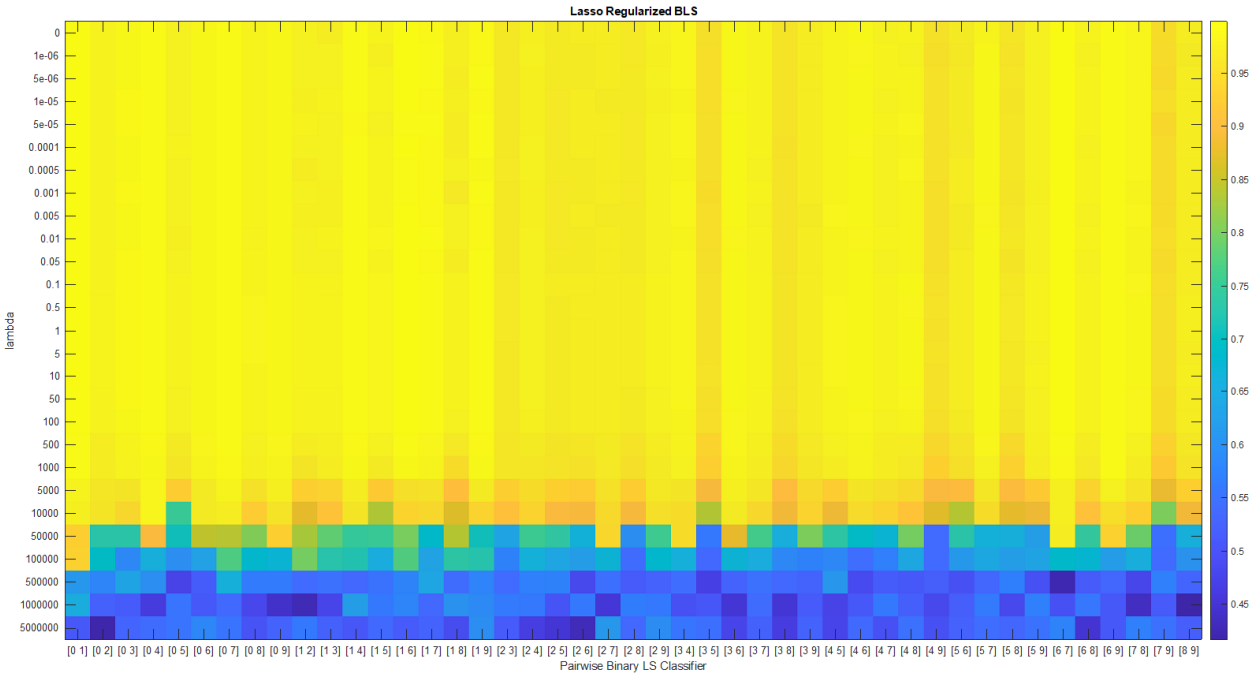
I have already done a lot of research into convolutional NN architecture and have a design in mind. The next steps are to complete the implementation of this design and synthesize what I have learned so far about k-NN and BLSC into a final report by the final deadline, Dec 12.

4.1 Link to Project

`https://github.com/twinfree/ECE532-Final-Project`



(a) Ridge



(b) LASSO

Figure 3: Validation accuracy for each pairwise classifier as a function of the parameter lambda.