Lab 6
CO902 – Probabilistic and statistical inference – 2012-13 Term 2
Lecturer: Tom Nichols
Data Reduction with Grayscale Handwritten Digit Identification:
Is Principal Components Analysis (PCA) data reduction better than variable selection?

Following on Lab 4, recall that the matlab file `digits.mat` has the 256 × 2000 `digit_img` matrix of pixel data (one digit per column), and the 2000-vector with true labels `digit_lab`. Split the data into training and test sets, $\mathbf{X}^{\text{train}}$ for the 256 × 1000 training sample and $\mathbf{X}^{\text{test}}$ for the 256 × 1000 test sample.

Consider different degrees of variable selection, reducing the data to $k_v < d = 256$ variable, and call this data $\mathbf{X}_v^{\text{train}}$ and $\mathbf{X}_v^{\text{test}}$. Recall from Lab 5 (see `Lab05.m` in the `Lab05_files.zip`) that we can rank the pixels by the variability of their class mean, and then consider only using the k_v most variable pixels. Be careful to only use $\mathbf{X}_v^{\text{train}}$ to determine which k_v pixels to use.

Consider different dimension PCA data reductions, reducing the data to $k_p < d$, and call this data $\mathbf{X}_p^{\text{train}}$. and $\mathbf{X}_p^{\text{test}}$. Again, use the *training data* to get the first k_p eigenvectors of the variance-covariance of the training data, call them $\mathbf{U}^{\text{train}}$. Specifically, the training data for Naïve Bayes is

$$\mathbf{X}_p^{\text{train}} = \mathbf{U}^{\text{train T}} \mathbf{X}^{\text{train}},$$

and each test case is then obtained with $\mathbf{x}_{pi} = \mathbf{U}^{\text{train T}} \mathbf{x}_i^{\text{test}}$, i.e. your complete test data matrix will be

$$\mathbf{X}_v^{\text{test}} = \mathbf{U}^{\text{train T}} \mathbf{X}^{\text{test}}.$$

These details are crucial… data reduction is a *preprocessing* step! Only the training data can be used to build **U** and choose pixels, and the data reduction must be applied identically to every case of training and test data[1].

Questions

1. Use $\mathbf{X}_v$ & $\mathbf{X}_p$ to find accuracy rates for different dimensions, at least $k_p = k_v = 1$, 5, 20, 50, 100 & 256. Make a plot comparing the accuracy rates for these two dimension reduction strategies. (Time permitting, consider more dimensions to make a more detailed comparison).

   Is there an optimal reduced data dimension? Or, a dimension below which no/little information is lost?

   If time. What if you had less training data? Reduce your training sample to 500 or 100 or 50 (keep the test sample size at 1000, to maintain good and comparable estimates of the out-of-sample prediction accuracy). Do your conclusions about optimal data reduction change?

2. *If time.* We previously found that optimal prediction was obtained using a fixed variance estimate over all voxels, all classes. Is this still the case with the reduced data?

Help

As the focus here is on PCA and data reduction, you are welcome to use my "solutions" from Lab5. See `Lab05_files.zip` for the functions `train_digit_classifier` and `test_digit_classifier`, as applied in the file `Lab05.m`.

[1] Note, in class, I applied PCA separately on each digit; that was just to demonstrate PCA, and not how you'd use it for data reduction for classification