ADS2 Problem Set 2.9:
Supervised learning: MNIST digit classification

Facilitators: GL, DS

April 16, 2024

We will continue and finalize feature extraction from MNIST handwritten digit dataset, now focusing on visualisation.

If you have not completed any part of the practical, please finish them first.

Once you are done, visualize means of your computed features for each digit class.

Now try to answer the following questions:

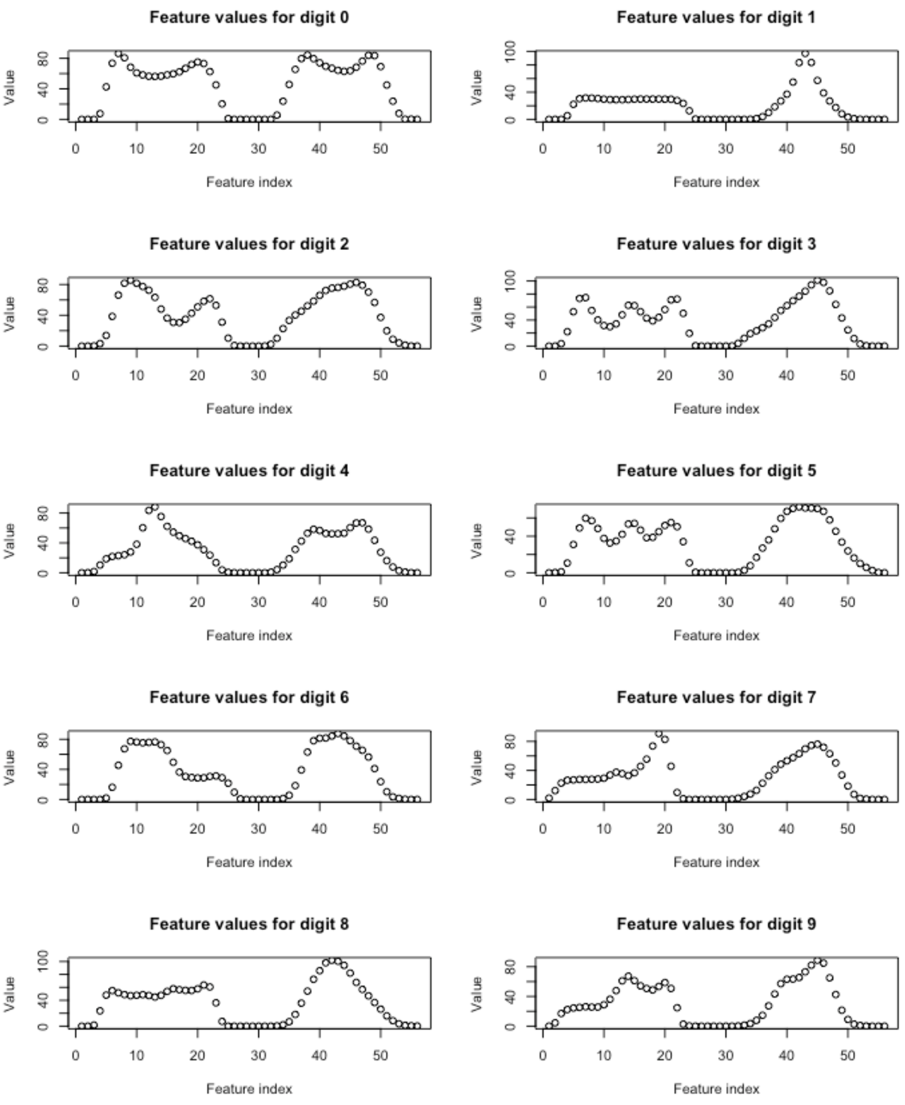
- Do you think your features would help the classification?
- Which digits might be hard to distinguish based on your features?
- Which might be easy?
- Could you classify some just by applying a threshold to values of certain features? (i.e. without a more sophisticated classifier like an artificial neural network).

First let's compute means for each label and feature:

```
fstats = matrix(1:560, nrow = 10, ncol = 56)
for (i in 1:10)
  for (j in 1:56)
     fstats[i,j] = mean(features[features$label==i-1,j+1]);
```

Now we can plot them, for example using the following code:

```
par(mfrow=c(5,2))
for (i in 1:10)
{ plot(fstats[i,], ylab = "Value", xlab = "Feature index")
  title(paste("Feature values for digit", toString(i-1)))
}
```



The first 28 features represent row means and the latter 28 column means.
Simple threshold-based classification may work for some digits, but not others.
For example, for digit 1, the sum of features seems clearly the lowest (uses the least ink).
For digit 7, feature 19 (“the upper dash”) is considerably higher than for all other digits.

For most others, it’s easy to find threshold-based separation rules that would work well against some, but not all digits. For example, for digit 6, feature 10 minus feature 20 would be higher than most, except perhaps 2 (as both have more ink at the bottom than the top). Similarly, for digit 3, feature 45 minus feature 40 would be higher than most, except perhaps 7 and 9 (all heavier on the right).

It’s important to note that we could not know how well such threshold-based separation would work by only looking at means across examples – we would have to plot distributions for this purpose. However, such preliminary analyses might nevertheless be useful in defining better features and subsequently improving classification or prediction performance, regardless of the training method used.

It is quite common in machine learning that using a fairly simple approach (like an MLP with a single hidden layer) based on good features would work better than using a very sophisticated approach without good features!