Jesse Pingitore

Machine Learning – Programming Assignment 02 Writeup

1. Implementation Notes

The key features of my implementation were: TODO. I found the most helpful matrix operations were numpy’s inverse, transpose, and dot functions, which saved me the time of having to do those operations via loops, and likely calculate them more efficiently as well.

The key performance bottlenecks is likely over pre-computing the mahalanobis distance matrix in MAP and Bayes. We need exponentially increase compute time and storage as the input data becomes larger, and as the number of possible classes to classify into becomes larger. It results in incredibly large models that cannot scale without, at least, parallelization for the compute issue.

1. MAP Classifier

I began by precomputing the distance matrix. This relies on my previous functions, especially the covariance and mahalanobis distance generator function being correct.

I failed to implement this, so I cannot comment further on implementation.

The reason that the decision boundary would more radically change from the linear classifier, and the KNN classifier, respectively, is: we are that we are considering more than just a linear regression with extra steps, and that we considering more than just mere next-neighbor distance, we are considering the deviation from the mean of the classifications, and balancing it with the overall likelihood of the class from prior experience. Both of these lead to more accurate wrapping of the decision regions.

1. Metal Parts

Unfortunately I have no bayes implementation. I cannot comment any further here as to implementation. I can say with certainty that that a full Bayes’ classifier would perform more satisfactory than a MAP, since it involves a manual weighting via a cost function. This lets us put artificial pressure on our data where we already knows exists in real life, to prevent the model from performing what is a mathematically sound operation, but produces an immensely undesirable and unapplicable result.

1. Scrap score plot (unequal costs)

[Even thought I didn’t complete this far, I know from class that we are going to experience anomalies in the Z axis.] The Z-axis is shaped the way is because it represents the cost. We can have massive regions be in the negative, because this indicates that we produce negative cost from the calculation – this profit, given the framing of sorting and recycling’s.