MSDS 422: Practical Machine Learning

Professor Anil D. Chaturvedi

Assignment 5: Principal Components Analysis

Tiffany Duong

Summary

Model validation is essential when it comes to machine learning, as it helps to verify that the model developed is performing as expected. Benchmarking is method of performance testing, which is core to model validation. In benchmarking, the model validator provides a comparison of the initial model being evaluated to another model or some type of metric. A way to benchmark would include having the model being used to evaluate against having a different method or methodology built into it compared to the initial model.

Research Design

For this assignment, we looked at the popular dataset from the MNIST database that is widely used in the field of machine learning. The MNIST dataset containing 70,000 examples of handwritten digits of 0-9. With the use of classifiers, we can separate the data into a training (60,000 examples) and test set (the remaining 10,000 examples) to see how well, or the probability the classifier model can successfully distinguish a handwritten digit into the correct associated number to it using 784 explanatory variables.

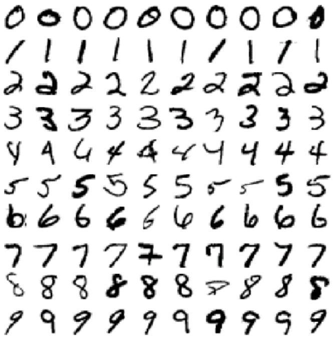

Figure 1: Sample images from the MNIST dataset

Technical Overview

To classify the examples in the dataset, we will make use of a random forest classifier, known as a powerful classifier, which fits a several decision tree classifiers and uses averaging to improve predictions and to control for overfitting. To validate the classifier model, it will be used as a benchmark for performance. Next, we will utilize another model consist of random forest classifier with principal component analysis (PCA) being applied before it. PCA is a dimensionality-reduction method used to transform a large set of features to a smaller set, while still retaining the specified amount of variability represented by the set of features (in this case variance=0.95). To assess model performance, the weighted F1-score - the harmonic mean of precision and recall - and the time it takes to execute these models will be used.

Results

For our benchmark model, we first applied a random forest classifier onto the full set of features and it had favorable performance with an F1 Score of 0.92 and taking 3.44 seconds to execute. For Model 2, we applied PCA to entire set of features to reduce the dimensionality from 700+ features, to only requiring 154 features to represent 95% of the variability in the dataset. This was put through the random forest classifier again and it resulted in a F1 Score of 0.89 and it took 7.97 seconds to execute. However, there was an issue with Model 2, in terms of code execution where only the training set needs to be fit and transformed, and the test set only being transformed, instead of the entire set being fit and transformed. So, for Model 3, which consisted of PCA and random forest, it resulted in better results than Model 2, which is a F1 Score of 0.91 (0.02 point increase) and taking 7.73 seconds to execute (3% reduction in execution time).

Overall, all three models performed favorable, however Model 3 that used PCA as a preliminary to random forest classification is recommended. When working with large datasets, it is a good idea to try to reduce the dimensionality of the dataset as much as possible, while still retaining as much information as possible (variability explained by the features). With having a lower dimensional dataset, it will be less prone to overfitting as well.