MODEL DEVELOPMENT OBJECTIVES

EXECUTIVE SUMMARY

In the domain of data science applications, we are constantly at odds with weighing the training time of model during the development phase, against the accuracy and reliability of our models results. In a frictionless environment, we would obviously always choose to be as accurate with our modeling approaches as possible; regardless of any other conflicting factors. With advances in cloud computing over the last decade we are giving less considerations to these run-time (or, train-time as it were) complexities, however, in currently the issue of dealing with finite resources is one we must contend with.

The realm of computer vision is an interdisciplinary field of research that combines some of the most complex areas of mathematics, computer science and engineering to design systems that are capable of automating tasks once thought only possible using the human visual system. Given the complexity of such systems, the amount of power needed to train and execute these systems is vast in terms of raw processor instructions, memory in the form of both random-access temporary store and long-term persistent storage.

The quintessential problem of computer vision is that of classifying hand-written digits systematically using a variety of algorithmic approaches. This problem is so common that it's often used as a benchmark of sorts for the performance of statistical learning classification systems. We will look to develop a modeling technique that not only solves this problem, it is performant in both its classification results and training-time.

RESEARCH DESIGN

For this problem we used a well-known, universally recognized and accepted database of hand-written digits composed of a series of 70,000, 28x28 pixelized, images that also contain an associated label that specifies the images true digit value. This database has been used in countless research studies, machine learning competitions (such as Kaggle), and academic classrooms throughout the world.

Our classification system uses the first sixty-thousand images in the database as a training dataset, and the last ten-thousand as our test and validation dataset. We will take the same sixty-thousand images and classify them with different approaches.

The first approach we will take is to establish our baseline results by using a standard random forest model on the trading set. This random forest will serve as our benchmark case throughout the rest of the study.

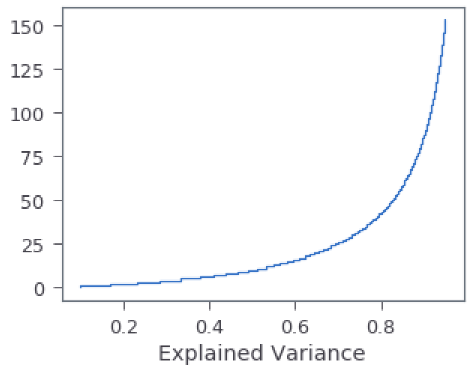
Once the benchmark case is solved for in terms of modeling accuracy and training-time, we will look to improve on it using a statistical procedure called principal component analysis. The big idea behind principal component analysis is to reduce the dimensionality of a dataset, making it easier to process, while preserving the variance in the data that makes each item uniquely identifiable. This technique is universally recognized as one of the most popular methods of optimization in this problem space.

TECHNICAL OVERVIEW

The model construction and benchmarking methods were conducted purely in the cloud, leveraging a pre-canned environment from a world-class provider of machine learning solutions, Google, called Colabratory. This cloud environment enables us to not only conduct research in a more efficient manner, while also allowing maximum reproducibility by taking out the variability of an individual desktops hardware configuration for the benchmark results. Additionally, we can push our research globally and allow any astute reader to reproduce our results for themselves in a sandboxed environment.

The process of benchmarking is conducted by creating our classifiers, then repeatedly fitting the model on the training data while timing the duration of the training event down to the millisecond.

For second part of the analysis, we will generate principal components that represent 95% of the variability in the explanatory variables in the dataset. We will take the same benchmarking process here, running it multiple times and storing our results for later analysis. The explained variance is additive, in that each principal component adds to the total explained variance in the data. We can see that for the MNIST dataset, using 154 principal components we can account for 95.03% of the varability in the data.

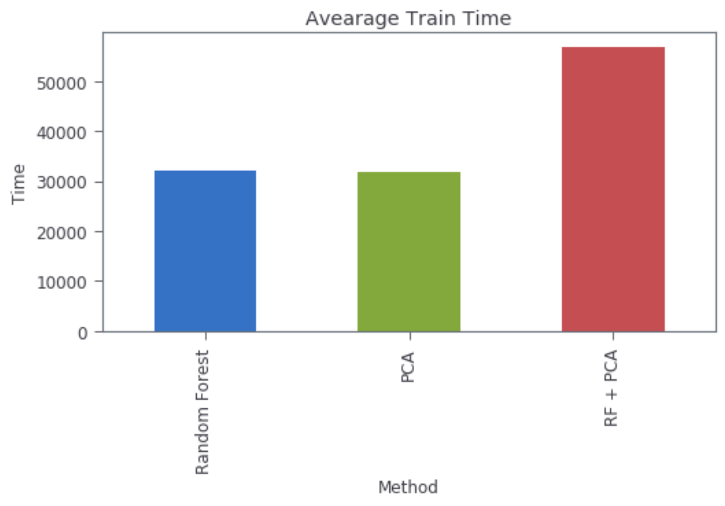


Finally, we will use another random forest classifier using the analyzed data from our principal component analysis. We will compare the performance of the approaches in terms of model accuracy, as well as training time.

CONCLUSION

After successfully completing our trials and research, we should first define how we measured each characteristic of the model. For accuracy, we used the F1 score (or, F-measure), which is a metric in statistical analysis that considers both the precision (true positives) and the recall (false positives). This score gives us a harmonic average of these metrics, on a scale of 0 to 1, where 1 is considered perfect.

The simple random forest produced a F-measure of .7642, without the use of any principal component analysis or any other boosting factors in the data. This is not a bad score, especially given that the average time to train the simple random forest was around 32k milliseconds.



For the sake of context, when we performed our principal component analysis alone, it took about the same amount of time as training the random forest model.

Unsurprisingly, the random forest trained on the data fitted with principal component analysis performed better in terms of accuracy. This model yielded a F-measure of .7726, a modest improvement of around eighty-four basis points.

As for model training time, the random forest combined with principal component analysis took an average of around 60k milliseconds, or an additional 47% time spent training the model.

The trade-off here essentially comes down to trading 84 basis points of precision for a 47% increase in training time. This dataset is extremely small in the universe of computer vision, weighing in at around 55 megabytes, which is tiny in comparison to some datasets in the gigabyte or even petabyte range. Since the gain in training time is well beyond significant, even with the cost of cloud computing falling, the estimated additional cost for hardware to train these models with principal component analysis would not be recommended at this time. Further research on other datasets and model configuration is recommended, as this analysis is in no way exhaustive.

For further information and details on this study was conducted, please visit the Colabratory Notebook.