Supervised Machine Learning in Various Dimensions

Derek So ddso@ucsd.edu

Abstract

In this paper, I evaluate three supervised machine learning methods on data with increasing levels of dimensionality, and different train/test cutoffs. I evaluate the performances of SVM, KNN, and Decision Trees on accuracy and record the effect of increasing dimensionality on the performance of the learning algorithms. Some of my findings may be confounded by inaccuracies in the datasets, but such can be accounted for in future experiments. Overall, I found that with increasing dimensionality came higher accuracy. Strangely, the Decision Tree model consistently performed better with less training, which is likely an abnormality and would not hold true elsewhere.

1. Introduction

Data is the ocean blue of our century—the sheer scale and availability of information along with our technical capabilities has and continues to exponentially increase. The internet has expanded and grown and with it our ability to categorize, store, and analyze information. Various experts have empirically evaluated the host of supervised learning methods in our current toolbox, both for relatively more rudimentary datasets with low dimensionality and more complex datasets with higher dimensionality (Caruana, R., Karampatziakis, N., & Yessenalina, A.). This paper also references other such studies with similar goals, namely STATLOG (King et al., 1995) and another experiment two years prior (Caruana & Niculescu-Mizil, 2006), going over their faults and limitations. In that spirit, I will be performing a similar experiment using some of the Methods they used on various datasets I acquired from Kaggle, of increasing complexity, and see if I can replicate their results in miniature.

2. Methods

A. Algorithms

I used three algorithms, support vector machines with radial basis functions (SVM (RBF)), k nearest neighbors (KNN), and decision trees (DT), all from the scikit learn python package. Their usage is broken down in the following:

SVM (RBF): Isolating two features, I repeatedly trained an SVM (RBF) classifier with a combination of C from the list [1, 10, 100, 100, 10000] and gamma from the list [1e-6, 1e-5, 1e-4, 1e-3, and 1e-2]. I isolate the classifier with the lowest training error and use that for testing. Due to hardware limitations, some of the training is cut to portions out of ten thousand rather than using the full original dataset. The other learning algorithms did not need such modifications.

KNN: I used a list of Ks from 1 to 9 and fit a classifier with a cv of 5, and took the classifier with the lowest training error and lowest k to use on the test dataset.

DT: Using the criterion of “entropy” and a random state of 1, I used depths from 1 to 5 to train several classifiers and opted for the classifier with the lowest training error and lowest depth to use on the test dataset.

B. Datasets

I acquired three different binary classification datasets from Kaggle to train and test the algorithms, explained in the following. Links to the source are in the references.

Music: A dataset of music files classified by their genre. Binary classification of pop or classical. I used the second csv of two hundred data poitns and removed some extraneous features, ending with seven, such as tempo and beats to train with.

LoL: A dataset of over fifty thousand recordings of the video game League of Legends. Binary classification was which of the two teams won. Thirty-plus features were reduced to seventeen major features, like game time and first ‘scores’.

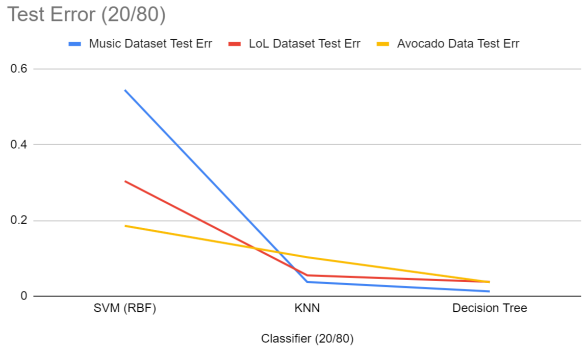
Avocados: A dataset of 18249 data points and 11 features about the sale of avocados. Binary classification of whether the avocados were organically or conventionally grown. Some features were price and volume.

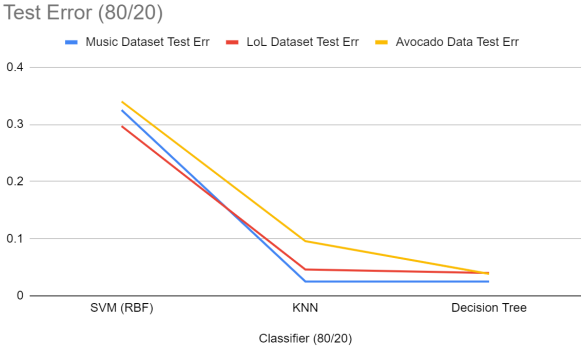
The performance of the algorithms is measured exclusively on the test error that resulted, in 80/20 and 20/80 partitions of the sourced datasets. As an example, the music dataset was split at 160/40 and 40/160. Again, due to technical limitations, 8000/2000 and 2000/8000 splits were used for SVM (RBF) training for the LoL and Avocados dataset.

3. Experiments

For each trained classifier for each partition for each dataset, a heatmap of training errors with respect to the algorithms or cross-validation errors was generated. A comparison of their accuracy on their respective test dataset is shown below. There was no notable difference in performance for different datasets. However, as we can see, the SVM (RBF) classifier consistently performed the worst, while both KNN and DT performing much better (DT negligibly better than KNN).

In regards to the different splits, classifiers trained on 80/20 splits generally performed better than classifiers trained on 20/80 splits, with 4 exceptions: SVM (RBF) on the Avocado dataset (.34 down to ~.19), and DT on all three datasets (.025, .04, ~.04 down to .0125, .038, .0368). On average, SVM (RBF) trained on 80/20 still performed better. It is possible for the occasional 20/80 train test partition to perform better than an 80/20 train test partition, as seen in the SVM (RBF) case. However, for such to occur for all three datasets for DT is a statistical anomaly, and my result is likely an aberration. The improvement is also all very small, 1.25% or less. Further explorations using other datasets will likely show that more training to less training creates a more accurate classifier.





Classifier (80/20)	Music Test Err	LoL Test Err	Avocados Test Err	Average Test Err
SVM (RBF)	.3250	.2970	.3400	0.3206666667
KNN	.0250	.0460	.0956	0.05553333333
DT	.0250	.0400	.0384	0.03446666667

Classifier (20/80)	Music Test Err	LoL Test Err	Avocados Test Err	Average Test Err
SVM (RBF)	.5440	.3036	.1856	0.3444
KNN	.0375	.0550	.1026	0.06503333333
DT	.0125	.0377	.0368	0.029

Example SVM (RBF) Heatmap for Music Dataset

4. Conclusions

There was no noteworthy difference in performance based on the different number of data points and number of dimensions between the chosen datasets. Instead, the difference may be found in the different types of algorithms and the different train test partitions. Overall, more training resulted in better classifiers. SVM (RBF) performed rather poorly, and another kernel likely would have been more optimal, while the KNN classifier performed surprisingly well. Like the previous empirical evaluation by Caruana, the decision tree classifier performed the best.

Acknowledgments

Thank you to Professor Zhuowen Tu and the TA staff who taught this course, especially for the forethought in their efforts in accommodating the COVID-19 crisis.

References

Caruana, R., Karampatziakis, N., & Yessenalina, A., (2008). An Empirical Evaluation of Supervised Learning in High Dimensions. ICML ‘08

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. ICML ’06, 161–168.

King, R., Feng, C., & Shutherland, A. (1995). Statlog: comparison of classification algorithms on large realworld problems. Applied Artificial Intelligence, 9, 259–287.

(Music) https://www.kaggle.com/insiyeah/musicfeatures

(LoL) https://www.kaggle.com/datasnaek/league-of-legends

(Avocado) https://www.kaggle.com/neuromusic/avocado-prices