ECE 759: Project Report

Phase 2

20th April, 2018

Submitted by,

Anupama Kesari
(akesari – 200199472)

Eshaan V Kirpal
(evkirpal - 200203773)

Introduction

In this phase of the project we perform cross validation and tuning of parameters for the classifiers built in the previous phase.

1. Cross-Validation

Cross-validation gives a measure of out-of-sample accuracy by averaging over several random partitions of the data into training and test samples. It can be used for hyperparameter tuning by doing cross-validation for several values of all parameters and choosing the parameter value that gives the highest accuracy. The process itself doesn't provide us with the parameter estimates, but it can be used to help make choices between alternatives.

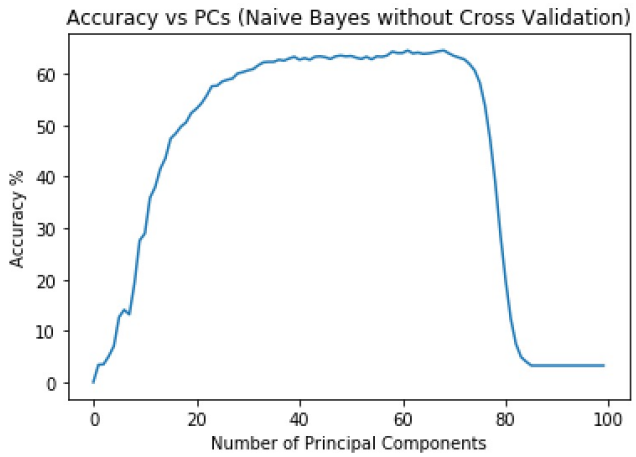
Tuning for Best Hyperparameters

Naïve Bayes

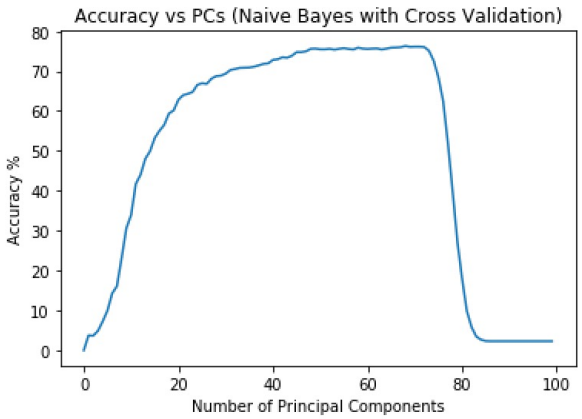
In the case of Naïve Bayes, there exists no parameters that may be tuned. The only change that maybe made is the distribution that is assumed for the attributes while finding the likelihoods. Since it was previously identified that the attributes are found to have a Gaussian-like distribution we can rule out the possibility of this correction. One check we may perform is the tuning to find the best number of Principal components to achieve highest accuracy. Comparing the performance with and without cross validation we find that,

YALE Dataset

Without Cross–validation –
Training set size = 1,207 images
Testing set size = 1,207 images

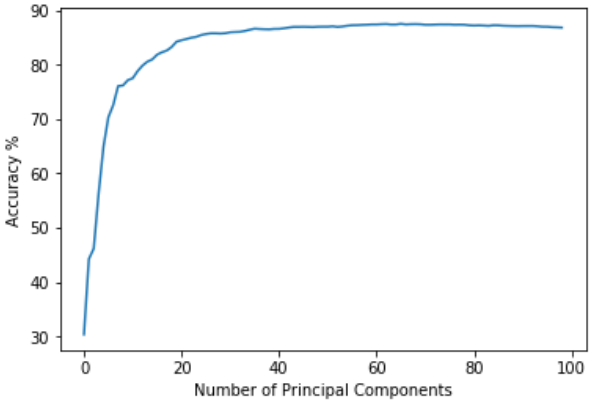


With Cross-validation the performance of the classifier changes as

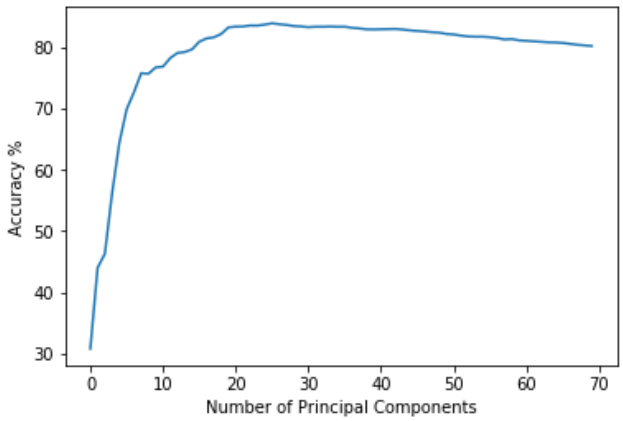


We see there is only a proportional increase in accuracy for the same number of PCs. We find that the best accuracy is achieved at 25 principal components. The same was observed prior cross-validation as well.

MNIST Dataset
Without Cross–validation –
Training set size = 35,000 images
Testing set size = 35,000 images



With Cross-validation the performance of the classifier changes as below



We see there is only a proportional increase in accuracy for the same number of PCs. We find that the best accuracy is achieved at 25 principal components. The same was observed prior cross-validation and hence there is no hyperparameter to be tuned for Naïve Bayes.

Decision tree

In case of Decision tree, it is found that pruning and bagging may help reduce overfitting the in the dataset. Since the algorithm above continually splits the branches until it can't reduce the entropy any further we stop splitting when the entropy is not reduced by a minimum amount. We tune this threshold value to get good accuracy. The accuracies before and after may be listed as

Before Cross-validation –

MNIST Dataset Results

Set of 2,000 training samples was considered.

No. of PCs considered	Accuracy
20	59.82 %
25	71.85 %
50	72.00 %
80	77.09 %

For 10,000 samples,

No. of PCs considered	Accuracy
25	79.83 %

(This took extremely long, >3 hours, to complete execution)

YALE B Dataset Results

Training set size = 1,207 images

Testing set size = 1,207 images

No. of PCs considered	Accuracy
20	35.02%
50	45.401%
80	45.89%
100	45.4%

Different Decision Tree Implementations: (With Cross-Validation) :

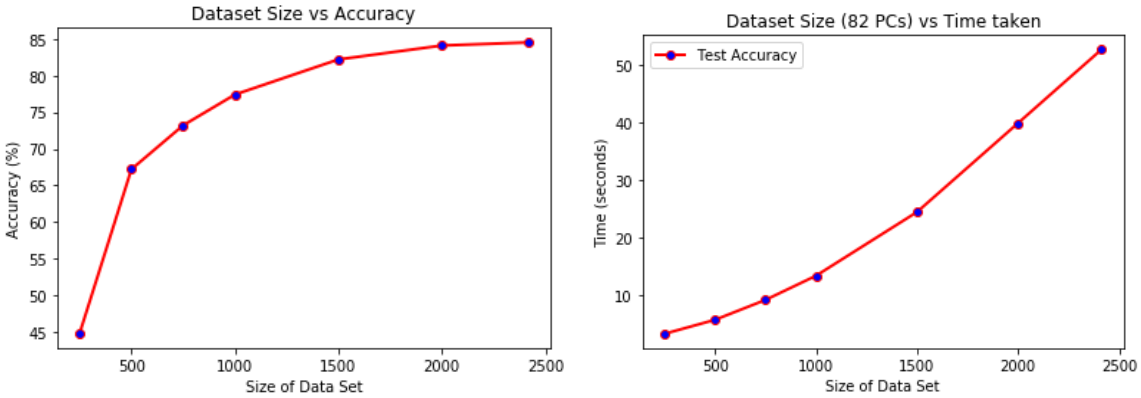
	MNIST	Yale B
Classical DT	77.09 %	45.89%
With Bagging (5- CV)	79.3%	48.9%
With Pruning (5- CV)	84.1%	61.61%

We observed that Bagging improves the performance of the classification for both the datasets. It was also observed that pruning improved the error rate.

2. Demonstration of Performances

Naïve Bayes on Yale Dataset –

The performance & time taken to perform the classification to the number of samples considered may be found as

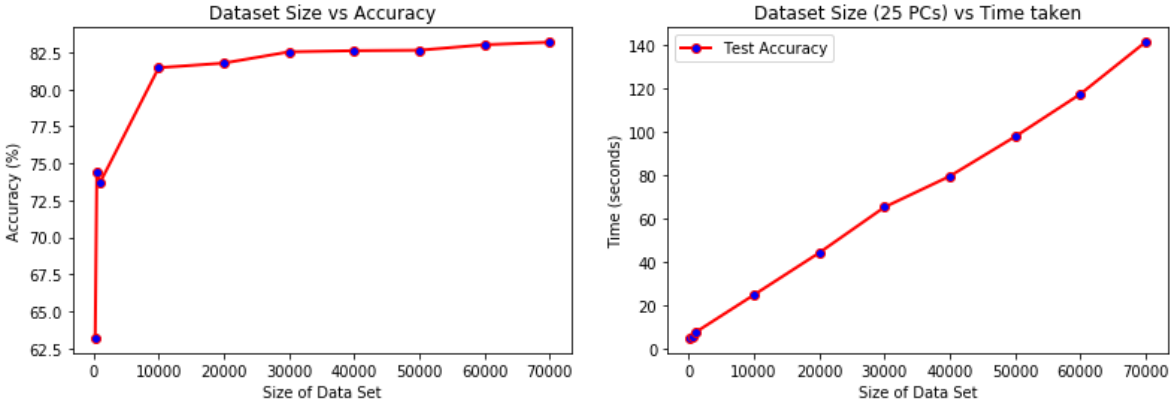


Performance with 5-fold Cross-validation

With an increase in number of samples the accuracy/performance of the classifier becomes steady and there is no further increase in accuracy. The time increases in an almost linear fashion.

In this case we have no Hyperparameters to tune.

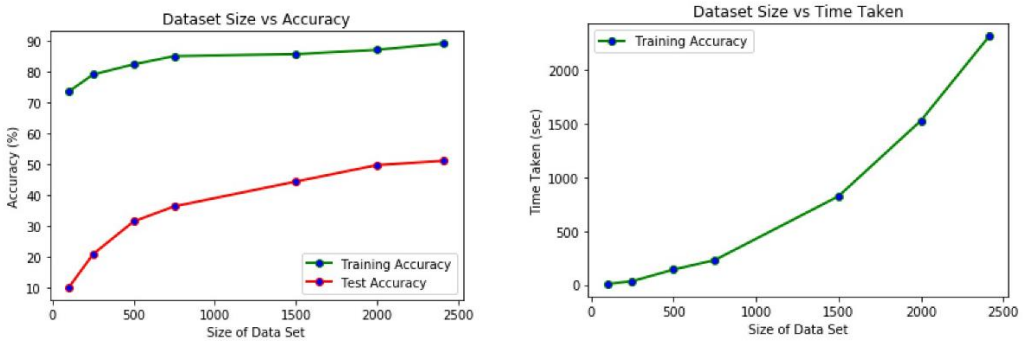
Naïve Bayes on MNIST Dataset –



Performance with 5-fold Cross-validation

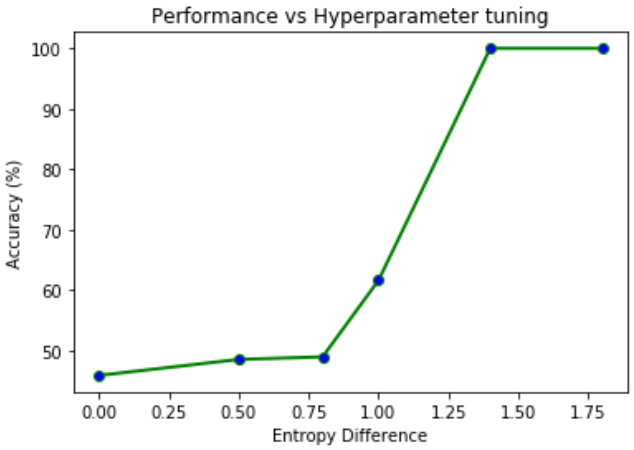
Similar to the performance on the Yale Dataset, with an increase in number of samples the accuracy/performance of the classifier becomes steady and there is no further increase in accuracy. The time increases in a linear fashion.

Decision Tree on Yale Dataset –



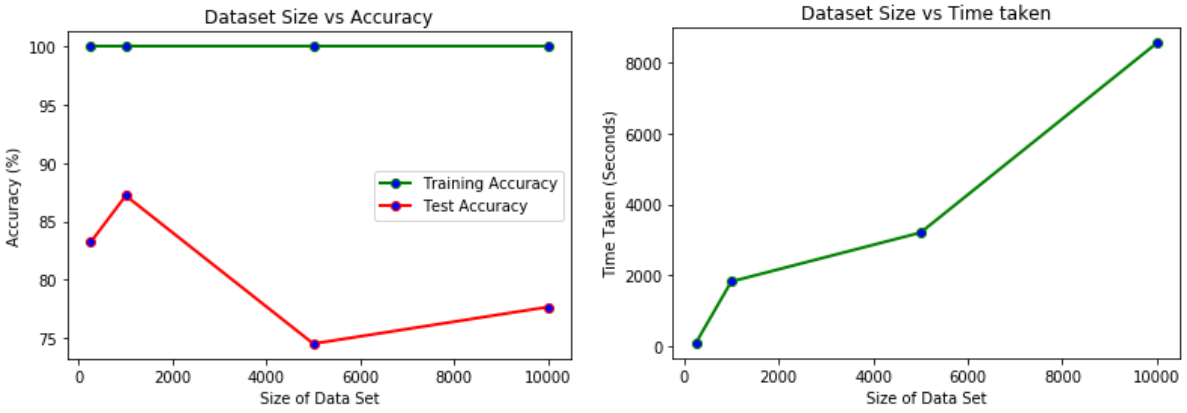
Performance with 5-fold Cross-validation

We find that there is a difference in the training and testing accuracies. This is because the number of sample images in each class is fairly low (32 for each class in test and train approximately) and the number of features is very high in comparison to the number of available samples. And for CV further division of the set to training and testing reduces the number of samples further.
The time again changes linearly.

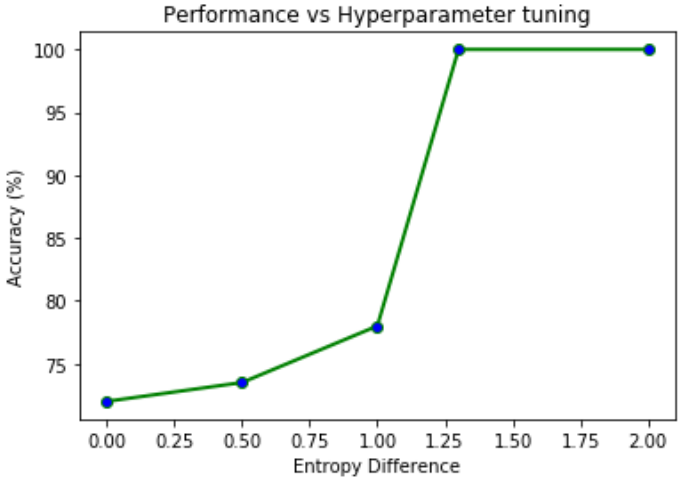


We find that the accuracy improves as the pruning threshold is modified. You can clearly see a growth/improvement in accuracy with the change in threshold.

Decision Tree on MNIST Dataset –



We find that the accuracy improves as the pruning threshold is modified. You can clearly see a growth/improvement in accuracy with the change in threshold.



We find that the accuracy improves as the pruning threshold is modified. You can clearly see a growth/improvement in accuracy with the change in threshold.

We find the training accuracy without pruning is at 100 since every sample is a leaf node.

3. Analysis of Results

a.
We find that with Naïve Bayes, if the data does not strictly follow any distribution then the likelihood value estimated by the assumption of some distribution will be wrong resulting in a bad estimate for the posterior probability which in turn results in bad classification with high error rate.

For Decision Trees, the attributes all having continuous values is an issue. This is because in order to find a split, the function takes very long to locate the best stump and thereby to build the tree. This issue increases with the increase of continuous-valued attributes and also with the number of samples. Another issue is the choice of number of branches at each node. It may be binary, ternary, etc.
In addition to this there is issue of having to evaluate and store the entire tree.

b.
For Naïve Bayes, the issue may be resolved by collecting data that is more suitably distributed so as to be able to estimate likelihood values better. Or by modifying the data so as to make it fit a distribution without drastically affecting the value/meaning of the data.

For Decision Trees, the continuous values issue may be resolved by group data together and assigning values or in a similar fashion converting them to categorical values. Another solution to this may be to take smaller sample sets and identify suitable stumps for every attribute and then extend the same stumps while considering the entire set.
The choice of number of branches may be resolved by identifying how the data is spread.
The storage issue may be resolved by performing pruning and retaining smaller trees.