Assignment 5

(ringale-rutakulk-smhaiska)

-

Which classifiers and which parameters would you recommend to a potential client?

→ Recommended Algorithm: Neural Net
Recommended Parameter Value: 20

We would like to recommend Neural Network classifier with hidden node count of 20.
This algorithm is faster and gives better results compared to both Nearest Neighbour and Adaboost algorithms. Accuracy given by Neural Net is about 74% for node count of 20.

How does performance vary depending on the training dataset size, i.e. if you use just a fraction of the training data?

Nearest Neighbour Algorithm:

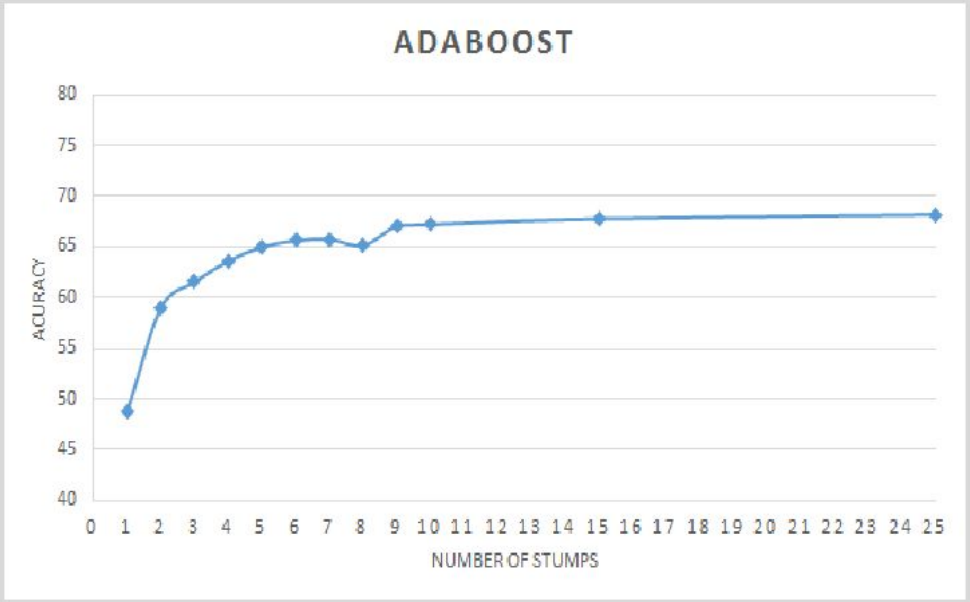
Data Size	Accuracy
100% (36000)	68%
60 % (21600)	66.17%
33% (10000)	67.23%

The algorithm takes longer time as the training set size is increased.

Adaboost:
For AdaBoost we have used 4000 feature pairs to construct the condition. These feature pairs are chosen randomly from total 192*192 pairs.
The AdaBoost algorithm worked best for 25 stumps with accuracy of ≈ 68%
With 4000 features, 25 stumps and entire training dataset, the total running time is ≈ 406 mins.
Following are the results for different training data size. It was obtained for 3 stumps and 4000 feature pairs.

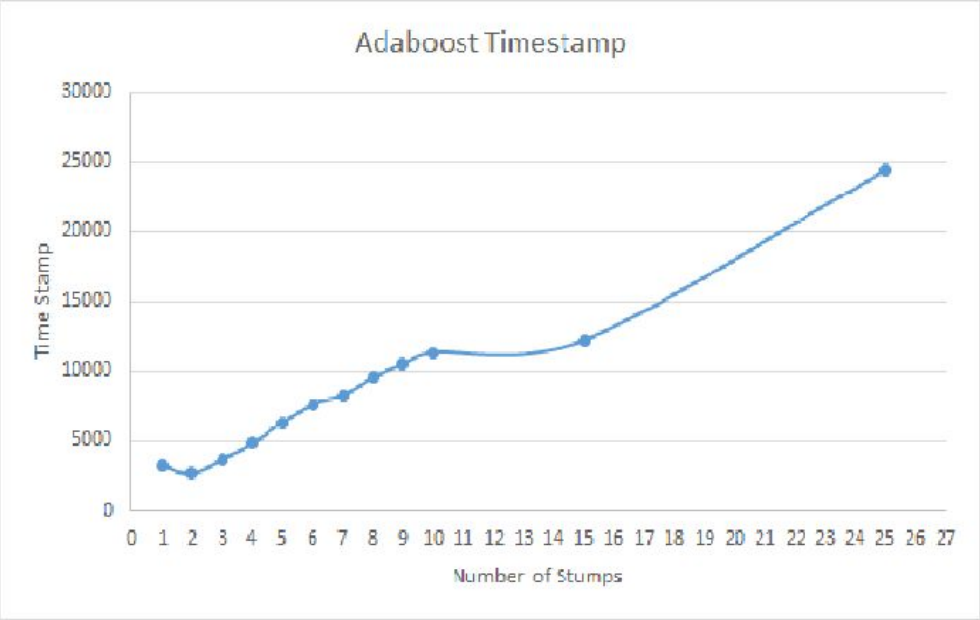
Data Size	Accuracy
100% (36000)	68%
50 % (28800)	65%
30 % (21600)	64%

The following graph shows accuracy of the algorithm for different number of stumps and 4000 feature pairs. The stumps that were considered were: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 25
These accuracy were obtained by taking average of 5 - fold cross validation.



Graph: Stumps vs Accuracy

The following graph shows the average time required for constructing and testing on different number of stumps. The time was recorded in seconds and is observed to increase with increase in number of stumps.



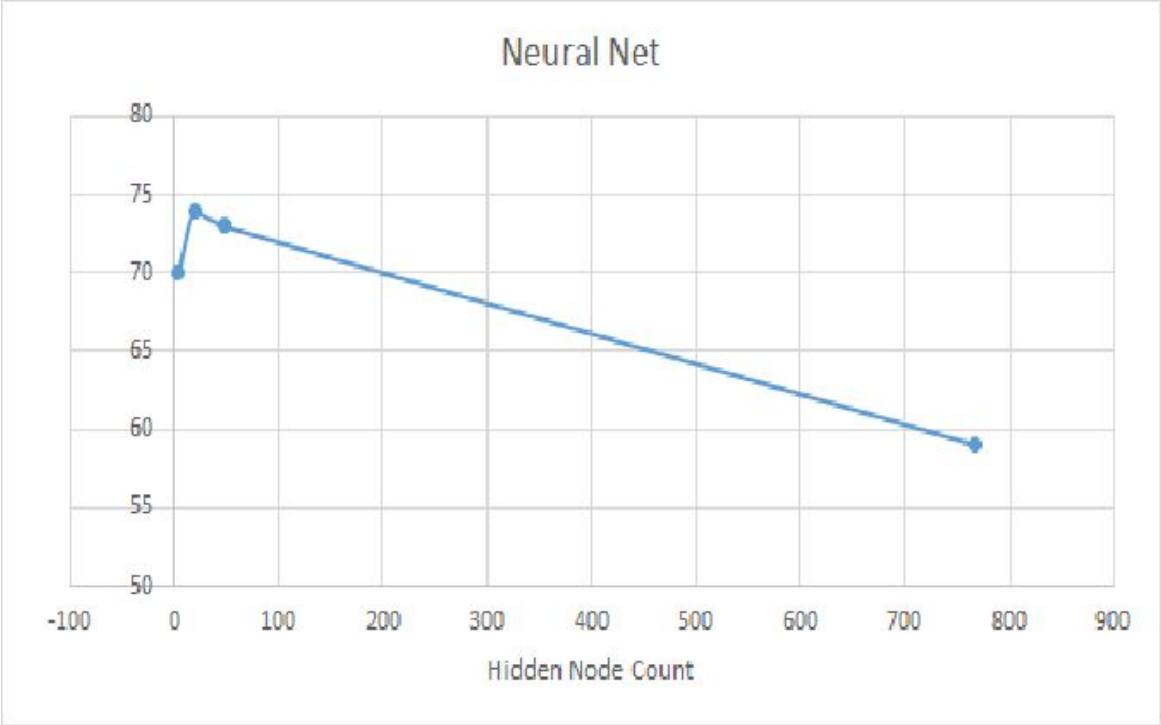
Graph: Stumps vs Timestamp

Neural Network:

We have used Stochastic Gradient Descent algorithm to update the weights since we have large amount of data and it is computationally faster compared to Batch Gradient Descent.

Data Size	Accuracy
100% (36000)	74%
33 % (10000)	68%

Using a fraction of the training data decreases the accuracy in case of Neural Network. Increasing the hidden node count increases the time taken for training the Neural Network



Graph: Hidden Node Count vs Accuracy

Show a few sample images that were classified correctly and incorrectly. Do you see any patterns to the errors?

Image 1

Image 2

Nearest Neighbour:
Image 1: 270
Image 2: 90

Nearest Neighbour algorithm predicted correctly for image 1 but gave a wrong prediction for image 2.

Adaboost:
Image 1: 270
Image 2: 90

Neural Network:
Image 1: 270
Image 2: 90

Neural Networks correctly classified the Image 1 and labelled it 270.
Neural Nets classified Image 2 incorrectly and given the label 90 instead of 0. May be because all the people are climbing the hill and might detect it as a vertical ridge assuming most of the ridges are horizontal.
Adaboost correctly classified the Image 1 and labelled it 270.
AdaBoost classified Image 2 incorrectly and gave it label 90 instead of 0. It might be because the condition feature that is chosen for split was one of those which had high contrast in horizontal images but here it's vertical since people are climbing a hill.