SML Assignment-4

Shobhit Raj (2022482)

1 Question 1

1.1 Approach

The task involved implementing AdaBoost.M1 algorithm on the MNIST dataset for binary classification of digits 0 and 1. The algorithm effectively utilized decision stumps as weak learners and PCA for dimensionality reduction. The validation accuracy plot helped in selecting the best boosted model, which was then evaluated on the test set to achieve the final accuracy.

1.1.1 Data Preparation and PCA Dimensionality Reduction

Samples belonging to classes 0 and 1 are filtered out from the training and testing sets. To create the problem suitable for classification, the labels 0 and 1 are relabeled as -1 and 1, respectively. The dataset is then split into training and validation sets, with 1000 samples from each class reserved for validation. The training dataset was then split into training and validation sets, ensuring an equal representation of both classes in the validation set. PCA is applied to reduce the dimensionality of the training set to 5 principal components. The reduced-dimension datasets are normalized by dividing by 255.

1.1.2 Decision Stump Learning

The core of the AdaBoost.M1 algorithm involves building a series of weak learners, known as decision stumps in this case. For each of the 5 principal components, midpoint splits were evaluated to find the best split that minimizes the weighted misclassification error. Predictions and majority values for each split were stored for later use in boosting iterations.

1.1.3 Boosting Iterations

In each boosting iteration, the algorithm computed weighted predictions based on the current weights assigned to each sample. It then identified the best split that minimized the weighted error and updated the weights accordingly. The update rule for the weights used the misclassification indicator and computed an alpha value, which serves as the weight for the current weak learner in the final ensemble.

1.1.4 Model Evaluation on Validation Set

For model evaluation, predictions were made on the validation set for each boosted model. The cumulative predictions were summed up, and the sign was taken to compute the final prediction for each sample. The accuracy on the validation set was then computed and stored for each boosted model to monitor the algorithm's performance over iterations.

1.2 Results

1.2.1 Validation Accuracy Analysis

The validation accuracy was plotted against the number of boosting iterations to visualize the algorithm's learning progress. This plot provided insights into how the accuracy improved with each iteration and helped identify the boosted model with the highest validation accuracy. The accuracy ranges from 99.55% to 99.7% for individual classes.

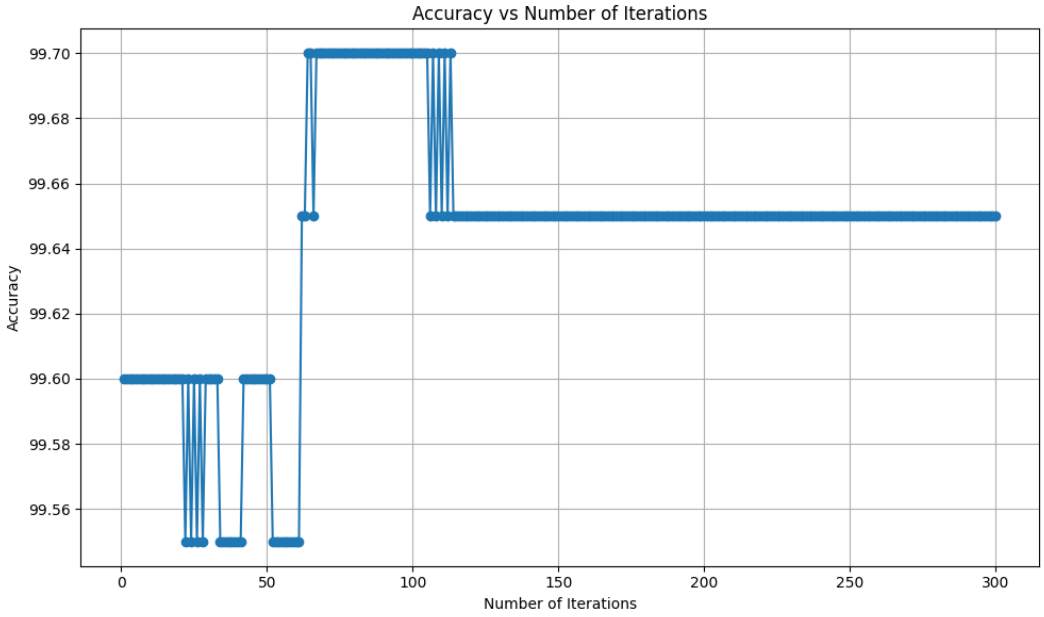

Figure 1: Accuracy vs Iterations plot

Maximum validation set accuracy was achieved at ***99.7%*** *with* ***113*** *iterations.*

1.2.2 Test Set Evaluation

Upon testing the selected boosted model on the test set, the final test accuracy was computed. This test accuracy served as the performance metric for the AdaBoost.M1 algorithm on the binary classification task of distinguishing between the digits 0 and 1 in the MNIST dataset.

Accuracy on Test Set of Boosted tree with Maximum Val Accuracy = **99.81087470449172%**.

2 Question 2

2.1 Assumptions

The relabelling of 0 as -1 is done in this question too, though it won't affect anything as its a regression problem.

2.2 Approach

The task was to apply gradient boosting with absolute loss to the regression problem. By iteratively training decision stumps to minimize SSR and updating residuals based on negative gradients, the algorithm effectively learned to approximate the target function. The validation MSE plot helped in selecting the optimal number of trees, while the test MSE provided a final assessment of the model's performance on unseen data.

2.2.1 Data Preparation and PCA Dimensionality Reduction

Samples belonging to classes 0 and 1 are filtered out from the training and testing sets. The labels 0 and 1 are relabeled as -1 and 1, respectively. The dataset is then split into training and validation sets, with 1000 samples from each class reserved for validation. The training dataset was then split into training and validation sets, ensuring an equal representation of both labels in the validation set. PCA is applied to reduce the dimensionality of the training set to 5 principal components. The reduced-dimension datasets are normalized by dividing by 255.

2.2.2 Decision Stump Learning with SSR

For each of the 5 principal components, unique values are identified and sorted in ascending order. The midpoints between consecutive unique values are evaluated as potential splits for the decision stumps. The best split is chosen to minimize the SSR, denoted as $h_t(x)$.

2.2.3 Gradient Update and Residue Computation

Residuals are updated using the formula $y - 0.01 \cdot h_t(x)$ where $h_t(x)$ is the prediction from the t-th decision stump. Labels for the gradient updates are computed as negative gradients.

2.2.4 Model Evaluation on Validation Set

After each iteration, the Mean Squared Error (MSE) is computed on the validation set to evaluate the model's performance. A plot of MSE against the number of trees provides insights into the learning progress of the model. The decision stump that yields the lowest MSE on the validation set is selected for evaluation on the test set.

2.3 Results

2.3.1 Validation MSE Analysis

The plot of MSE against the number of trees provided insights into how the model's performance improved with additional trees. The decision stump that results in the lowest MSE on the validation set is selected for further evaluation.

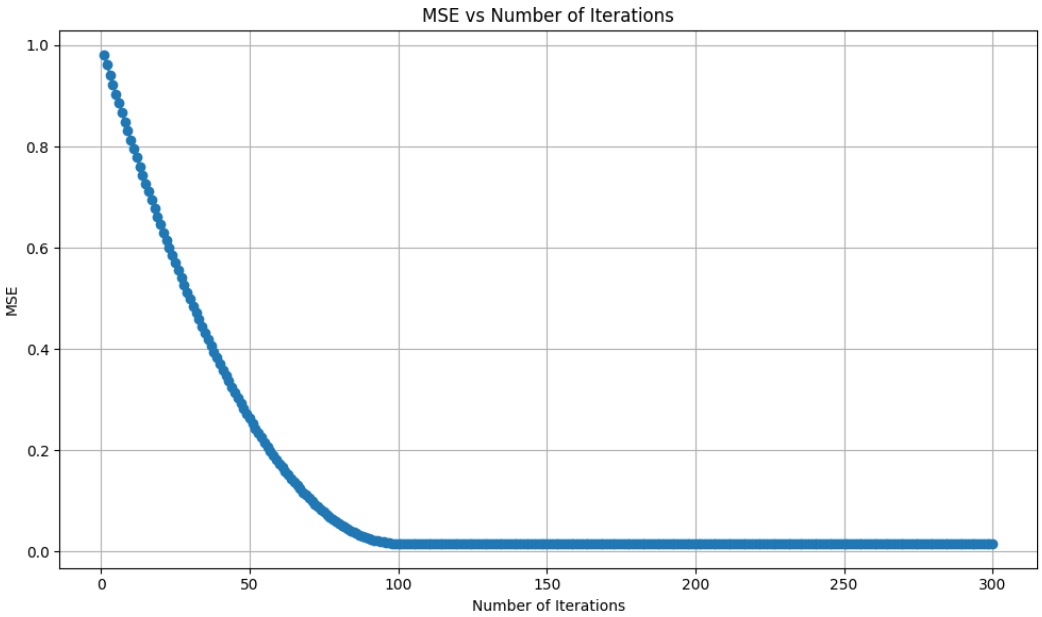

Figure 2: MSE vs Iterations plot

Minimum validation set MSE was achieved at ***0.015764408980526466*** *with* ***299*** *iterations.*

2.3.2 Test Set Evaluation

The decision stump that yielded the lowest MSE on the validation set was evaluated on the test set. The MSE between the predicted and actual values on the test set was computed to assess the model's generalization ability.

MSE on Test Set of Boosted tree with Minimum Val MSE= **0.015028905316582113**.