Multi-Class Adaboost: Methodology and Results

Introduction

Adaboost is an ensemble learning technique whereby multiple weak classifiers are combined to produce a strong classifier. The Multi-Class Adaboost implementation is an extension of this technique in handling multi-class classification problems. This report is going to show the methodology followed, model training, validation results, and a model selection strategy.

Methodology

Multi-Class Perceptron as Weak Learner

In the code, weak learners are Multi-Class Perceptrons. A perceptron is the simplest linear classifier that iteratively updates the weights with respect to its prediction errors. For a multi-class classification, the perceptron is generalized to learn a binary classifier for each class, encoded in one-vs-rest mode. Each class's binary classifier predicts whether a given instance belongs to that class.

Training of the perceptron involves:

- Initializing the weights and bias for each class.
- Iterative weight updating on misclassifications based on the learning rate specified by a user α.
- Early termination in the case of class-specific classifier accuracy exceeding the weak threshold.

Multi-Class Adaboost Framework

In Adaboost, weak learners are trained in a series to then be assigned weights according to their performance and combined into one robust ensemble classifier. The major steps involved in the process of Multi-Class Adaboost include the following:

- Initialization: Assign uniform weights to all training samples.
- Weak Learner Training: Train a weak learner-a Multi-Class Perceptron-on the weighted dataset.
- Error Calculation: Calculate the weighted error of the weak learner.
- If the error is greater than 50%, the weak learner is rejected.
- Alpha Calculation: Calculate the weight of the weak learner using the formula :
  w new=w old+α·y·X
  where α α is the learning rate, y is the target label, and X is the feature vector.

- Weight Update: Update sample weights to emphasize the misclassified samples.
- Normalization: Normalize sample weights.
- Prediction Aggregation: Aggregate weak learner predictions by weighted majority vote.
- By iterating these steps, Adaboost focuses more on difficult samples, enabling the ensemble to handle complex decision boundaries effectively.

Model Training

Dataset and Preprocessing

The model was trained and evaluated on the Digits dataset from the sklearn library. This dataset consists of 8x8 grayscale images of handwritten digits (0-9) with 64 features (pixel intensities) and corresponding labels.

Steps involved in preprocessing:

- Splitting the dataset into training (80%) and test sets (20%), stratified by class to maintain the balance of classes.
- Feature Scaling: The features were standardized, using StandardScaler to have equal ranges for all features, improving convergence during training.

Training Process

Multi-Class Adaboost model initialization with:

- Weak Learner: Multi-Class Perceptron.
- Number of Estimators: 50 (the maximum number of weak learners).

The process of training consists of:

- Training 50 weak learners in a sequence such that each subsequent weak learner focuses on those samples misclassified by the previous ones.
- The computing weighted error of every weak learner, adjusting the sample weight to emphasize the harder-to-classify samples.
- Saving each weak learner with the alpha value regarding it.

Results and Validation

Model Evaluation

The test set was used for testing the model on basic metrics in classification problems:

- Accuracy: the overall correct proportion
- Classification report: class-wise precision, recall, F1 score of each class
- Confusion matrix: Actual vs Predicted Label for the purpose of analyzing the errors.

These are as follows.

- Accuracy: 96.25% on the test set, which is a very strong overall performance.
- Classification Report: High precision and recall across most classes, minor misclassifications in visually similar digits, such as 3 vs. 8.
- Confusion Matrix: Most of the diagonal elements were high to confirm the robustness of the model, though there were some off-diagonal errors reflecting class confusions.

Model Selection Strategy

The following are considered as strategies for selecting the best model:

- Validation Accuracy: The model with higher validation accuracy was preferred.
- Error Analysis: Some of the important insights from the confusion matrix were that there is a need to emphasize the specific misclassification of particular digits.
- Early Stopping: Weak learners were discarded during training whose weighted error exceeded 50% to avoid overfitting.

Contributions

We worked together throughout the entire project, discussing and making decisions collaboratively at each stage. We regularly exchanged ideas, considered different solutions, and ensured we were aligned in our approach. This collaborative approach allowed us to complete the project effectively and in a balanced manner

We would like to acknowledge the assistance of ChatGPT in providing guidance and suggestions during the implementation

Conclusion

The Multi-Class Adaboost showed very high accuracy and robustness on the Digits dataset, effectively leveraging weak learners to form a strong classifier. By focusing on misclassified samples during training, it successfully tackled class imbalance and complex decision boundaries. This showcases the power of ensemble methods in improving classification performance across truly diverse datasets.