Objective Document: Implementation and Evaluation of Adaboost Algorithm for Non-Linear Classification

Overview

The objective of this task is to implement and evaluate the Adaboost algorithm for solving a non-linear classification problem using the circles dataset. The task involves the development of a custom Adaboost implementation, experimentation with various weak classifiers, and visualization of results to assess performance.

Dataset Description

The dataset used for this task is a synthetic, non-linear dataset (circles dataset) generated using the following Python code:

```
from sklearn.datasets import make_moons, make_circles
import matplotlib.colors as colors
import matplotlib.pyplot as plt
X, y = make_circles(n_samples=500, noise=0.1, random_state=42, factor=0.2)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
plt.scatter(X[:,0], X[:,1], c=y, cmap=colors.ListedColormap(["blue", "red"]))
plt.axis('equal')
plt.show()
```

The generated dataset will visually resemble two concentric circles. This dataset is inherently non-linear and requires advanced classification techniques for effective separation.

Task 1: Implementation of Adaboost Algorithm

Objective

Develop a custom implementation of the Adaboost algorithm from scratch to classify the non-linear dataset.

Specifications

1. **Input Data**:
   - Samples: $$ x_1, x_2, \ldots, x_n $$
   - Labels: $$ y_1, y_2, \ldots, y_n $$, where $$ y \in \{-1, 1\} $$
2. **Initialization**:
   - Assign initial weights $$ w_{i,1} = \frac{1}{n} $$ for all samples.
3. **Algorithm Steps**:
   - For each iteration $$ t = 1, 2, \ldots, T $$:

- Select a weak learner $$ h_t(x) $$ that minimizes the weighted error:
  $$
  \epsilon_t = \sum_{i=1}^n w_{i,t} \cdot I(h_t(x_i) \neq y_i)
  $$
- Compute the weight of the weak learner:
  $$
  \alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)
  $$
- Update sample weights:
  $$
  w_{i,t+1} = w_{i,t} e^{-\alpha_t y_i h_t(x_i)}
  $$
- Normalize weights so that $$ \sum w_{i,t+1} = 1 $$.

4. **Final Classifier**:
   Combine weak learners into an ensemble classifier:
   $$
   F_T(x) = \sum_{t=1}^T \alpha_t h_t(x)
   $$
5. **Hyperparameter**:
   Introduce a hyperparameter $$ \eta $$ (default value: 0.5) to control learning rate.

Deliverables

- Implementation of the algorithm.
- Visualizations:
  - Classifier fit at each iteration.
  - Final decision boundary of the ensembled classifier.

Task 2: Evaluation with Weak Classifiers

Objective

Evaluate the performance of the custom Adaboost implementation using various weak classifiers and tune hyperparameters to achieve optimal performance.

Weak Classifiers

The following weak classifiers will be used:

- Logistic Regression (LogReg)
- Decision Stump
- Decision Tree (depth=3)
- Linear Support Vector Machine (SVM)
- Linear Discriminant Analysis (LDA)

Procedure

1. Train the Adaboost algorithm with each weak classifier.
2. Tune hyperparameters for both Adaboost and the weak classifiers to maximize classification accuracy.
3. Generate visualizations for each model:
    - Left plot: Classifier fit at every iteration.
    - Right plot: Final ensembled decision boundary.

Performance Target

Achieve classification accuracy greater than 98% based on data visualization and model evaluation.

Expected Outputs

1. Custom implementation of the Adaboost algorithm.
2. Visualizations demonstrating iterative classifier fits and final decision boundaries for each weak classifier.
3. Performance metrics (e.g., accuracy) for all models.

This document outlines the tasks and objectives in a structured manner to guide the implementation and evaluation process effectively.

Sources
[1] Instructions-Assignment-8.pdf
https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/53601675/4cf4c17c-5678-4394-b507-eaf24e6f1cb2/Instructions-Assignment-8.pdf
[2] DA5401 Assignment 8
https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/53601675/4cf4c17c-5678-4394-b507-eaf24e6f1cb2/Instructions-Assignment-8.pdf