Adaptive Debiased Lasso in High-Dimensional GLMs with Streaming Data (ADL)

This repository contains the source code for the **Approximated Debiased Lasso (ADL)** algorithm, designed for online statistical inference in high-dimensional generalized linear models (GLMs) with streaming data. The algorithm is particularly useful for scenarios where data arrives sequentially, and efficient, real-time inference is required.

Repository Structure

The repository is organized as follows:

Deomstrations for replicating Numerical Results

In the first demonstration, we choose $n = 200, p = 500$ and $s_0 = 6$. We consider two cases of different covariance matrix $\mathbf{\Sigma} = 0.1 \times \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$ and $\mathbf{\Sigma} = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$. We construct confidence intervals for three randomly selected parameters from each category of $\boldsymbol{\beta}^*$. See Section 4 for more details of the simulation settings.

- [run_case1.py]: Execution file for replicating results of Table 1.
- [run_case2.py]: Execution file for replicating results of Table 2.

In the second demonstration, We increase the value of p to 20000 and choose $n = 1000, s_0 = 20$ with $\mathbf{\Sigma} = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$. We construct confidence intervals for three randomly selected parameters from each category of $\boldsymbol{\beta}^*$.

- [run_case3.py]: Execution file for replicating results of Figure 3.

Real Data Example

- [run_realdata.py]: Execution file for real data analysis with online data feeding.

Algorithm Core Functions

- [radar.py]: Contains built-in functions for **Online Regularization Annealed Epoch Dual Averaging (RADAR)** and **Adaptive RADAR**.
- [adl.py]: Implements the **Approximated Debiased Lasso (ADL)** algorithm.
- [adl_realdata.py]: Implements the ADL algorithm for real data analysis, compatible with sparse arrays.

Helper Functions

- [cal.py]: Contains helper functions and a data generator for simulations.
- [process.py]: Handles data processing tasks, such as extracting uni-gram and bi-gram features from raw data and converting them into sparse matrices.

Dependencies

To use the ADL package, ensure your Python environment has the following dependencies installed:

- `numpy`
- `scipy`
- `matplotlib`
- `pandas`

Usage

Replicating Numerical Results

To run the simulations included in this repository, execute the following files:

- For **Simulation 1**:

  ```
  python run_case1.py
  python run_case2.py
  ```

- For **Simulation 2**:

  ```
  python run_case3.py
  ```

Real Data Example

Dataset

For real data analysis, the dataset `combined_data.csv` is required. This dataset can be downloaded from:

- Email Spam Classification Dataset on Kaggle

Make sure to place the dataset in the root directory of the repository before running the real data analysis scripts.

To extract uni-gram and bi-gram features from raw data:

```
python process.py
```

To run the real data analysis:

```
python run_realdata.py
```