README: Implementation for *"Addressing both variable selection and misclassified responses with parametric and semiparametric methods"*

Hui Guo, Grace Y. Yi[*], and Boyu Wang

Overview

This repository contains the implementation of the methods proposed in the manuscript submitted to *Bernoulli*, titled:

"Addressing both variable selection and misclassified responses with parametric and semiparametric methods"
by Hui Guo, Grace Y. Yi [*], and Boyu Wang.

The code includes both parametric and semiparametric methods for addressing response misclassification and performing variable selection in binary classification settings.

Directory Structure

- `train/`: Core implementations
  - `param.py`: Parametric method implementation
  - `semi.py`: Semiparametric method implementation
  - `path_following.py`: Approximate path-following (APF) method from Liu and Zhang (2014)
  - `test.py`: Evaluation script
  - `utils.py`: Supporting utilities
- `run.py`: Entry point for running the full method pipeline.
- `example.ipynb`: Jupyter notebook demonstrating:
  - Synthetic data simulation
  - End-to-end usage of the proposed methods

Usage

You can invoke the main functionality via `run.py`, specifying inputs and configuration arguments as needed.

[*]corresponding author.

Required Data Inputs

- `Z`: Main-study covariate matrix.
  - `numpy array` of shape (n, p)
- `Y_star`: Noisy binary responses in main-study data.
  - `numpy array` of shape $(n,)$
- `Z_val`: Covariates in validation data.
  - `numpy array` of shape (n_v, p)
- `Y_val`: True labels in validation data.
  - `numpy array` of shape $(n_v,)$
- `Y_star_val`: Noisy labels in validation data.
  - `numpy array` of shape $(n_v,)$
- `discrete_idx`: List of indices for discrete features.
  - `list`
  - Each element takes values in $\{0, 1, ..., p-1\}$
  - default: []
- `Z_test`: Covariates in test data. Optional.
  - `numpy array` of shape (n_{test}, p)
  - default: `None`
- `Y_test`: True labels in test data. Optional.
  - `numpy array` of shape $(n_{test},)$
  - default: `None`
- `test`: Boolean flag indicating whether to test.
  - `bool`
  - `True` or `False`
  - default: `False`

Model Setup

- `link_func`: Link function.
  - `str`
  - `'logit'` or `'probit'`
  - default: `'logit'`
- `penalty`: Penalty type.
  - `str`
  - `'l1'`, `'scad'`, or `'mcp'`
  - default: `'scad'`
- `use_intercept`: Boolean flag indicating whether to include intercept.

  - `bool`
  - `True` or `False`
  - default: `True`

- `criterion`: Model selection criterion.
  - `str`
  - '`gcv`' or '`bic`'
  - default: '`gcv`'

- `model_running`: Method type.
  - `str`
  - '`param`' or '`semi`'
  - default: '`semi`'

- `densityType`: Density estimation method (for semiparametric method).
  - `str`
  - '`Kernel`' and '`pcaKernel`'
  - default: '`pcaKernel`'

Hyperparameters

- `eta`: Decreasing coefficient for the sequence of regularization parameters.
  - `float` in $[0.9, 1)$
  - default: 0.91

- `R`: Projection radius.
  - positive `float`
  - default: 0.91
  - If set to `None`, the algorithm computes it automatically

- `L`: Initial learning rate.
  - small positive `float`
  - default: 0.05

- `N_iter`: Number of iterations (outer loop).
  - `int`
  - default: 5

- `max_loop`: Maximum number of loops.
  - `int`
  - default: 20