

CODE README OF SUBMISSION #448

Anonymous authors

Paper under double-blind review

This file delineates the contents of the `Implementation` folder found in the supplementary material of submission #448.

1 ARCHIVE CONTENTS

The `datasets` directory houses the specific data structures and the raw caches of the datasets used in the experiments. The file `adjunct_func.py` implements some auxiliary functions, while `algorithms.py` provides the implementation for the proposed algorithm. The script `data_rounds.py` is employed for the organization of data in each online round during the experimentation phase. Refer to `environment.txt` for details on environmental prerequisites, specifically noting that the `Scipy` version required must be 1.4.4 or higher.

2 NOTATIONS

To facilitate understanding, Table 1 elucidates the primary identifiers in the implementation, correlating them with the mathematical symbols used in the main paper.

Table 1: Correspondence between identifiers and their respective symbols.

Identifier	Notation
<code>[X_init, y_init]</code>	X°
<code>[X_del, y_del]</code>	X^-
<code>[X_add, y_add]</code>	X^+
<code>[X_val, y_val]</code>	$\{\tilde{\mathbf{x}}_i, \tilde{y}_i\}_{i=1}^N$
<code>lam1</code>	λ_1
<code>lam2</code>	λ_2
<code>b0</code>	$\hat{\beta}^{(i)}$
<code>active</code>	\mathcal{A}
<code>inactive</code>	$\mathcal{S}_{\mathcal{Z}}$
<code>bt_A</code>	$\hat{\beta}_{\mathcal{A}}$
<code>X_init_A</code>	$\{\mathbf{x}_{i,\mathcal{A}} \mid \forall i \in [1, n_0]\}$
<code>X_del_A</code>	$\{\mathbf{x}_{i,\mathcal{A}} \mid \forall i \in [n_0+1, n_0+n_-]\}$
<code>X_add_A</code>	$\{\mathbf{x}_{i,\mathcal{A}} \mid \forall i \in [n_0+n_-+1, n_0+n_-+n_+]\}$
<code>bt_inA</code>	$\hat{\beta}_{\mathcal{S}_{\mathcal{Z}}}$
<code>omegas</code>	$\omega_i \ (i \in \mathcal{S}_{\mathcal{Z}})$
<code>mu</code>	$\mu(\cdot)$
<code>nu</code>	$\nu(\cdot)$
<code>key_mat</code>	\mathcal{L}
<code>X_init_inA</code>	$\{\mathbf{x}_{i,\mathcal{S}_{\mathcal{Z}}} \mid \forall i \in [1, n_0]\}$
<code>X_del_inA</code>	$\{\mathbf{x}_{i,\mathcal{S}_{\mathcal{Z}}} \mid \forall i \in [n_0+1, n_0+n_-]\}$
<code>X_add_inA</code>	$\{\mathbf{x}_{i,\mathcal{S}_{\mathcal{Z}}} \mid \forall i \in [n_0+n_-+1, n_0+n_-+n_+]\}$

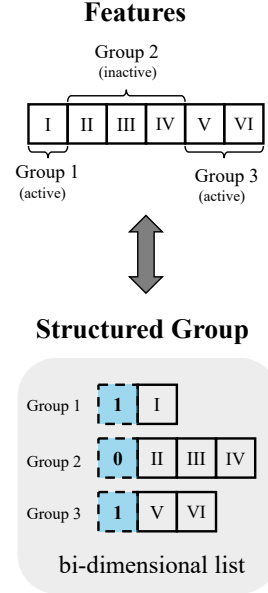


Figure 1: Structured data design for group-wise penalization, preserving both shrinkage and grouping information.

3 REMARKS ON GROUP-WISE REGULARIZER

For efficient organization, we employ a specialized data structure to manage the active group, exemplified in Figure 1, which diverges from the original definition of \mathcal{A} . The partition here is stored as a bi-dimensional Python list, with each sublist's first value serving as a flag bit. This list should be manually configured prior to its integration into the `SAGO` framework.