2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 1-4, 2023, Oahu, Hawaii, USA

Group Lasso with Checkpoints Selection for Biological Data Regression

1st Huixin Zhan
Department of Computer Science
Texas Tech University
Lubbock, Texas, USA.
https://orcid.org/0000-0001-8926-1941

2nd Yifan Wang
Department of Mathematics and Statistics
Texas Tech University
Lubbock, Texas, USA,
https://orcid.org/0000-0003-3292-4972

2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 979-8-3503-3702-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/SMC53992.2023.10394634

***Abstract*—Some unique characteristics of biological data are (1) that they are always High-Dimension and Low-Sample-Size (HDLSS) and (2) there are changes in the data distribution, such as an imbalance in classes, distribution and covariate shifts, etc. In this paper, we propose a Group Lasso with Checkpoints SElection (GL_CSE) algorithm to tackle both issues. To address the first issue, we utilize a group Lasso regression model tailored for HDLSS data to perform feature selection on predefined groups of features, alleviating overfitting and being invariant under group-wise orthogonal reparameterizations. To address the second issue, we propose the checkpoint selection method to extract important model checkpoints while training on group Lasso via two proposed metrics, i.e., the average KL-divergence between training and validation features and the Frobenius error of the covariance matrices between training and validation features. Both metrics aim to select model checkpoints with minimal drifts between the training and validation features. The results of our experiments indicate that our proposed GL_CSE algorithm achieves better performance compared to other baseline methods in terms of the MSE and R^2 measurements. Specifically, on the biological age dataset, our GL_CSE method achieves 0.8799 and 0.9883 for the MSE and R^2 measurements, respectively. Additionally, we also show that our proposed checkpoint selection method performs better than regular K-fold cross-validation. Specifically, on the biological age dataset, GL_CSE (Q2) achieves 0.9045 MSE and 0.9880 R^2, respectively, which outperforms the regular K-fold cross-validation results, i.e., 1.0612 MSE and 0.9871 R^2, respectively.**

***Index Terms*—HDLSS, biological data, group Lasso regression, checkpoint**

I. Introduction

High-dimensional, low-sample-size (HDLSS) data refers to datasets that have a large number of variables/features (high-dimensional) but a limited number of observations/examples (low-sample-size). The problem with traditional machine learning methods, such as logistic regression, discriminant analysis, and k-nearest neighbors, is that they often encounter difficulties when handling HDLSS data [1]. These challenges include overfitting, low prediction accuracy, and computational inefficiencies.

To overcome these challenges, researchers have developed several techniques for analyzing HDLSS data. One approach is to use methods that do not involve dimensionality reduction (DR), such as regularized regression, decision trees, and support vector machines. These techniques work by explicitly modeling the relationships between the features and the target variable, without reducing the dimensionality of the data. For instance, Shen et al. [14] proposed a no-separated data maximum dispersion classifier that finds a projecting direction that maximizes the interval in which all training samples scatter. Gunduz and Fokoue [1] compared the predictive performances of robust classification techniques with a special concentration on robust discriminant analysis. However, this category of approaches trains on high-dimensional features, resulting in a large model size and time-consuming training that is often impractical. Additionally, specialized analysis techniques are required to address the unique characteristics of biological data regression. For example, in biostatistics, datasets often have features grouped by biological pathways, which require specialized analysis techniques. Therefore, when conducting biological data regression, it's essential to assess the structural identifiability of models to determine whether model parameters can be uniquely determined *prior to regression* and what data are required to achieve that. Consequently, we propose a DR based method named Group Lasso with Checkpoints SElection (GL_CSE) algorithm. Another main approach involves using DR techniques as a preprocessing step before regression, including methods such as principal component analysis (PCA) [10], linear discriminant analysis (LDA) [17], t-distributed stochastic neighbor embedding (t-SNE) [15], and others. These techniques work by transforming the original high-dimensional data into a lower-dimensional space that retains the most significant features while minimizing information loss. Some examples of methods in this category are DR based on feature selection [5] (e.g., L1-penalized logistic regression [3] and HSIC-Lasso [16]), transformation-based projection [7], manifold-based approaches [11], regularity [8], ensemble learning [13], neural networks [12], and multi-view learning [9]. By reducing the data's dimensionality, these techniques can enhance traditional machine learning models' performance while reducing computational complexity, making it feasible to train a small model and deploy it on devices with limited resources. For instance, the maximum margin criterion (MMC) [4] was

proposed as an improvement over linear discriminant analysis (LDA) for high-dimensional, low-sample-size (HDLSS) datasets by avoiding the need to solve the inverse of a low-rank between-class scatter matrix. However, this approach may suffer from degradation in classification performance due to data piling.

Over the past decade, biological data regression has achieved breakthroughs in applications, e.g., pre-regression and post-regression diagnostics, with large sample size [2]. However, when facing high dimension, low sample size (HDLSS) data, such as predicting biological age [1] using DNA methylation data, regression models face two main challenges: **1)** Sparsity: HDLSS data often contains a large set of features but only a few number of samples, resulting in a sparse dataset. This sparsity can make it infeasible to find meaningful patterns in the data. **2)** Drift in Data Distribution: Biological data can exhibit changes in the data distribution over time, causing an imbalance in classes, domain shifts, and other issues that make it difficult to generalize the regression model to new data.

Recently, in order to address the data sparsity problem, Li et al. [6] proposed a logistic regression with adaptive sparse group lasso penalty (LR-ASGL) to simultaneously perform cancer diagnosis and adaptive gene selection. However, applications of penalized regression models should be further extended to a group setting, because developing generic regression models for a wider range of HDLSS biological data is desired. Therefore, our study focuses on HDLSS data regression in a broad biological domain, e.g., biological age prediction, acute myeloid leukemia (AML)/acute lymphoblastic leukemia (ALL) identification, lung disease identification, and B-cell chronic lymphocytic leukemia (B-CLL) identification. To tackle the data sparsity problem, we utilize a group Lasso regression model specifically designed for HDLSS biological data to perform feature selection on predefined feature groups, leading to improved regression performance.

To address the data drifting problem, we propose the GL_CSE algorithm. This algorithm selects important model checkpoints while training on the group Lasso via two novel metrics: the average KL-divergence between training and validation features and the Frobenius error of the covariance matrices between training and validation features. Both metrics encourage the selection of model checkpoints with a more slight drift between training and validation features. Specifically, the first metric ensures the selection of checkpoints with low KL-divergences between training and validation features. The second metric ensures the selection of checkpoints with smaller Frobenius errors of the covariance matrices between training and validation features.

Our contributions are as follows: **1)** Our proposed GL_CSE algorithm provides generic effective regression models on a wide range of HDLSS biological data, e.g., biological age prediction, AML_ALL identification, lung disease identification, and B-CLL identification. **2)** To address the two aforementioned challenges, we propose the GL_CSE algorithm, which selects essential model checkpoints while training on the group Lasso using two metrics: (1) the average KL-divergence between training and validation features and (2) the Frobenius error of the covariance matrices between training and validation features. Both metrics promote the selection of model checkpoints with minimal drift between training and validation features.

[1] https://www.kaggle.com/competitions/aging

II. Method

In this section, we will first introduce group Lasso regression and the two proposed metrics, i.e., average KL-divergence between training and validation features (ϵ_{KL}) and the Frobenius error of the covariance matrices between training and validation features (ϵ_{F}). We will then introduce the proposed GL_CSE algorithm.

A. Group Lasso Regression

Given $\mathcal{D}$ samples $\mathcal{X} = \{x_1, x_2, ..., x_\mathcal{D}\}$ and the labels $\mathcal{Y} = \{y_1, y_2, ..., y_\mathcal{D}\}$, $x_i \in \mathbb{R}^N$ and $y_i \in \mathbb{R}^1$, we first split the data into training set, test set, and validation set with ratios $r_{\text{train}} = \frac{|\mathcal{X}_{\text{train}}|}{|\mathcal{X}|}$, $r_{\text{test}} = \frac{|\mathcal{X}_{\text{test}}|}{|\mathcal{X}|}$, $r_{\text{val}} = \frac{|\mathcal{X}_{\text{val}}|}{|\mathcal{X}|}$, to obtain $\mathcal{X}_{\text{train}}$, $\mathcal{Y}_{\text{train}}$, $\mathcal{X}_{\text{test}}$, $\mathcal{Y}_{\text{test}}$, $\mathcal{X}_{\text{val}}$ and $\mathcal{Y}_{\text{val}}$.

The goal of a Lasso regression model [3] is to find the value of the parameters β that minimizes the sum of squared errors as follows:

$$\min_{\beta \in \mathbb{R}^N} \left\{ \|\mathcal{Y} - \mathcal{X}\beta\|_2^2 + \lambda \|\beta\|_1 \right\}. \tag{1}$$

A large λ value places greater importance on the penalization, resulting in more zero parameters among the β values. This is particularly useful in high-dimensional data, where there are more features than observations, but only a small fraction of the features are expected to significantly contribute to regression performance. However, in some cases, the features in $\mathcal{X}$ naturally have a grouped structure. For example, in biostatistics, the datasets often have features grouped by patient characteristics. Lasso regression provides individual sparse solutions, rather than group sparse solutions. Therefore, we will use group Lasso regression for biological data to promote sparsity within feature groups.

Thus, assume the N features are divided into $\mathcal{G}$ groups, with N_g indicating the number in group g, we use a matrix $\mathcal{X}_g$ to represent the matrix of features of the g-th group, with corresponding parameter vector β_g. The group Lasso can be solved as follows [6]:

$$\min_{\beta \in \mathbb{R}^N} \left\{ \left\| \mathcal{Y} - \sum_{g=1}^{\mathcal{G}} \mathcal{X}_g \beta_g \right\|_2^2 + \lambda \sum_{g=1}^{\mathcal{G}} \sqrt{N_g} \|\beta_g\|_2 \right\}, \tag{2}$$

where the $\sqrt{N_g}$ terms accounts for the varying group sizes, and $\|\cdot\|_2$ is the Euclidean norm (not squared). The group Lasso regression applies a similar procedure as Lasso, but at the group level, where an entire group of features may be excluded from the model based on the value of λ.

B. Average KL-Divergence (ϵ_{KL})

Each feature $x_{i,n}$ in $\mathcal{X}_{\text{train}}$ and $\mathcal{X}_{\text{val}}$ has a sample space $\mathcal{X}_n$. For probability distributions $\mathcal{P}_{n,\text{train}}$ and $\mathcal{P}_{n,\text{val}}$ defined on $\mathcal{X}_n$, the KL-divergence between $\mathcal{P}_{n,\text{train}}$ and $\mathcal{P}_{n,\text{val}}$ is defined as:

$$\text{KLD}_n(\mathcal{P}_{n,\text{train}} \parallel \mathcal{P}_{n,\text{val}}) = - \sum_{x_{i,n} \in \mathcal{X}_n} \mathcal{P}_{n,\text{train}}(x_{i,n}) \log \left(\frac{\mathcal{P}_{n,\text{val}}(x_{i,n})}{\mathcal{P}_{n,\text{train}}(x_{i,n})} \right). \tag{3}$$

Thus, the ϵ_{KL} for all N features can be computed as follows:

$$\epsilon_{\text{KL}} = \frac{1}{N} \sum_{n=1}^{N} \text{KLD}_n. \tag{4}$$

C. Frobenius Error of the Covariance Matrices (ϵ_F)

The covariance matrix $\text{CM}_{\mathcal{X}_{\text{train}}}$ of $\mathcal{X}_{\text{train}}$ can be computed as the matrix whose (i, j) entry is the covariance as follows:

$$\text{CM}_{\mathcal{X}_{i,\text{train}} \mathcal{X}_{j,\text{train}}} = \text{E}[(x_{i,i} - \text{E}[x_{i,i}])(x_{i,j} - \text{E}[x_{i,j}])], \tag{5}$$

where the operator E denotes the mean value. The covariance matrix $\text{CM}_{\mathcal{X}_{\text{val}}}$ of $\mathcal{X}_{\text{val}}$ can be computed in the same way. Thus, the ϵ_{F} can be computed as:

$$\epsilon_F = \frac{\|\text{CM}_{\mathcal{X}_{\text{train}}} - \text{CM}_{\mathcal{X}_{\text{val}}}\|_F}{\|\text{CM}_{\mathcal{X}_{\text{train}}}\|_F}. \tag{6}$$

D. The Proposed GL_CSE Algorithm

In order to encourage the selection of model checkpoints with minimal drifts between the training and validation features, we propose a GL_CSE algorithm to extract important model checkpoints while training on group Lasso via the above two proposed metrics. During the training process, we set the thresholds Γ_{KL} and Γ_{F} for ϵ_{KL} and ϵ_{F}, respectively, we will only select checkpoints $\mathcal{C}$ that satisfy both $\epsilon_{\text{KL}} \leq \Gamma_{\text{KL}}$ and $\epsilon_{\text{F}} \leq \Gamma_{\text{F}}$. During the training process, the first metric ϵ_{KL} ensures the selection of checkpoints with low KL-divergences between training and validation features, while the second metric ϵ_{F} ensures the selection of checkpoints with smaller Frobenius errors of the covariance matrices between training and validation features. The pseudocode is shown in Algorithm 1. For each episode i, we will first randomly shuffle the data $\mathcal{X}$ with the labels $\mathcal{Y}$ and split the dataset into K folds. For each fold j, we will then split $\mathcal{X}_{\text{train}}$, $\mathcal{X}_{\text{test}}$, and $\mathcal{X}_{\text{val}}$ in Line 5. If both thresholds are satisfied, we will save the checkpoint $\mathcal{C}^{i,j}$ for the i-th episode and j-th fold into the selected checkpoints set $\mathcal{C}^*$. It should be noted that while group Lasso is used, the computation of the two metrics we proposed is done by considering all features with equal parameters.

Algorithm 1 The GL_CSE Algorithm

Require: Episode number E, the fold number K, and $\mathcal{C}^* = \{\emptyset\}$.
Ensure: Checkpoints $\mathcal{C}^*$.
1: **for** $i \leftarrow 1$ to E **do**
2: Randomly shuffle the data $\mathcal{X}$ with the labels $\mathcal{Y}$.
3: Split the dataset into K folds.
4: **for** $j \leftarrow 1$ to K **do**
5: Take the j-th fold as $\mathcal{X}_{\text{test}}$, take the $(j+1)\%K^{th}$ fold as $\mathcal{X}_{\text{val}}$, and take the remaining folds as $\mathcal{X}_{\text{train}}$.
6: **if** $\epsilon_{\text{KL}} \leq \Gamma_{\text{KL}}$ (in Equation 4) and $\epsilon_{\text{F}} \leq \Gamma_{\text{F}}$ (in Equation 6) **then**
7: Solve $\hat{\beta}$ using the group Lasso in Equation 2 on $\mathcal{X}_{\text{train}}$.
8: $\mathcal{C}^{i,j} = \hat{\beta}$
9: $\mathcal{C}^* = \mathcal{C}^* \cup \mathcal{C}^{i,j}$
10: **end if**
11: **end for**
12: **end for**

III. Evaluation

In this section, we will discuss the experimental settings, datasets, baselines, evaluation measurements, and experimental results obtained from the GL_CSE algorithm.

A. Settings

All datasets considered are split into a training set, a test set, and a validation set with ratios $r_{\text{train}} = 0.8$, $r_{\text{test}} = 0.1$, and $r_{\text{val}} = 0.1$. We set the fold number $K = 10$ and the total episode number $E = 50$. Penalization parameter λ is selected by minimizing 5-fold cross-validated error. Thresholds Γ_{KL} and Γ_{F} are set as the first quartile (Q1) and the second quartile (Q2) of all ϵ_{KL} and ϵ_{F} values, respectively. Principle component analysis (PCA) [10] is used to project the original high-dimensional features to a matrix of principal components. The number of the projected features is selected to maintain an explained variance of 95%. This step is implemented via the built-in modules in Scikit-learn [2].

B. Datasets

The following real-biological HDLSS datasets are considered in our experiments: **1) Biological age dataset** [3] in which contains 100 DNA Methylation samples in 100 classes. Every sample consists of 483756 features. **2) ALL_AML dataset** [4] in which comprises 72 samples, divided into two classes: ALL and AML. The class ALL has 47 samples, while AML has 25 samples. Every sample contains 7129 values for gene expression. **3) LUNG dataset** [5] contains a total of 203 samples across 5 classes: adenocarcinomas, squamous cell lung

[2] https://scikit-learn.org/stable/
[3] https://www.kaggle.com/competitions/aging
[4] https://www.kaggle.com/datasets/crawford/gene-expression
[5] https://jundongl.github.io/scikit-feature/datasets.html

carcinomas, pulmonary carcinoids, small-cell lung carcinomas, and normal lung. The number of samples for each class is 139, 21, 20, 6, and 17, respectively. Each sample consists of 3312 features. **4) CLL_SUB_111 dataset** [6] consists of 111 samples of gene expressions obtained from high-density oligonucleotide arrays. The dataset contains three classes, which represent genetically and clinically distinct subgroups of B-CLL. Every sample consists of 11340 features.

The data statistics are summarized in Table I.

TABLE I: Statistics for real-world datasets.

Data	# of Samples	# of Features	# of Classes
Biological age	100	483756	100
ALL_AML	72	7129	2
LUNG	203	3312	5
CLL_SUB_111	111	11340	3

C. Baselines

To evaluate the performance of our method, we compare our method with the following two popular regression baselines: **1) L1-penalized logistic regression [3]** ($\log \mathcal{R}$-L1). This method performs linear feature selection with the L1 regularization for regression problems. **2) HSIC-Lasso [16]**. This method is a nonlinear method that uses kernels to learn a sparse model on kernelized labels and features. It is also recognized as the state-of-the-art minimum redundancy maximum relevance (mRMR) model.

D. Evaluation Metrics

We will compare our GL_CSE with the above two approaches, i.e., $\log \mathcal{R}$-L1 and HSIC-Lasso, using the two evaluation measurements as follows: **1)** The mean squared error (MSE) is a measure of the average squared difference between the predicted values and the true values. It is calculated using the following formula:

$$\text{MSE} = \frac{1}{\mathcal{D}_{\text{test}}} \sum_{i=1}^{\mathcal{D}_{\text{test}}} (y_i - \hat{y_i})^2, \tag{7}$$

where y_i is the true value of the i^{th} label, $\hat{y_i}$ is the predicted value of the i^{th} label, and $\mathcal{D}_{\text{test}}$ represents the number of samples in the test set.

2) The R-squared (R^2) value is a statistical measure that represents the proportion of the variance in the dependent variable that is explained by the independent variable(s). It is calculated using the following formula:

$$\text{R}^2 = 1 - \frac{\text{SS}_{\text{res}}}{\text{SS}_{\text{tot}}}, \tag{8}$$

where SS_{res} is the sum of squares of residuals (the difference between the true labels and the predicted labels), which can be computed as: $\text{SS}_{\text{res}} = \sum_{i=1}^{\mathcal{D}_{\text{test}}} (y_i - \hat{y_i})^2$.

[6]https://jundongl.github.io/scikit-feature/datasets.html

SS_{tot} is the total sum of squares (the difference between the true labels and the mean of the true labels), and it is computed as: $\text{SS}_{\text{tot}} = \sum_{i=1}^{\mathcal{D}_{\text{test}}} (y_i - \bar{y_i})^2$, where $\bar{y_i}$ is the mean of the true labels.

E. Experimental Results

Our experiments address two main questions: **1)** Does the GL_CSE method show a better performance in biological data regression compared to the other popular regression methods? **2)** Is our checkpoint selection strategy better than the other checkpoint selection methods, such as K-fold cross-validation?

We will address these questions in subsection III-E1 and subsection III-E2. In subsection III-E3, we will conduct an ablation study that demonstrates the importance of the proposed checkpoints selection module in our GL_CSE.

1) Performances: In Table II, we will report the experimental results of our GL_CSE compared with two baselines, i.e., $\log \mathcal{R}$-L1 and HSIC-Lasso using MSE and R^2 measurements. In this experiment, Γ_{KL} and Γ_{F} are set as the Q1 of all ϵ_{KL} and ϵ_{F} values, respectively.

TABLE II: Experimental Results on MSE and R^2.

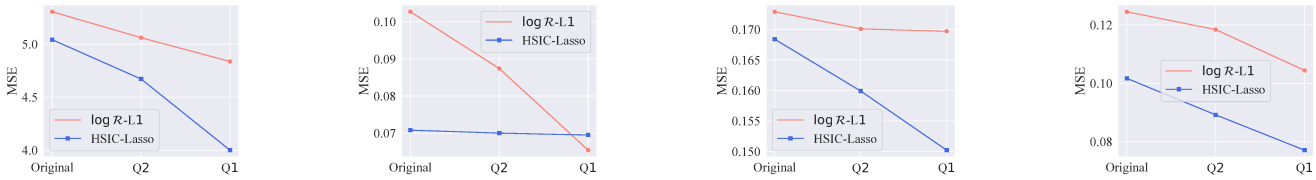
		Biological Age	ALL_AML	LUNG	CLL_SUB_111
$\log \mathcal{R}$-L1	MSE	5.3042	0.1027	0.1729	0.1245
	R^2	0.8009	0.9371	0.9131	0.9014
HSIC-Lasso	MSE	5.0412	0.0708	0.1684	0.1017
	R^2	0.8204	0.9593	0.9294	0.9175
GL_CSE (Q1)	MSE	**0.8799**	**0.0316**	**0.1192**	**0.0847**
	R^2	**0.9883**	**0.9991**	**0.9889**	**0.9899**

The MSE reflects the average difference between the predicted labels and the true labels. Therefore, a lower MSE value indicates better predictive power. In contrast, the R^2 value ranges from 0 to 1, where 1 indicates a perfect fit of the model to the data, and 0 indicates that the model does not explain any variability in the data. Therefore, a higher R^2 value is desired. In Table II, we highlight the top performing outcomes in bold. It is noteworthy that across all the datasets, our proposed GL_CSE method demonstrates superior performance compared to other methods, as evidenced by both measurements. The HSIC-Lasso method ranks second, followed by the $\log \mathcal{R}$-L1 method. Specifically, in the case of the biological age dataset, our GL_CSE achieves outstanding results, with MSE and R^2 values of 0.8799 and 0.9883, respectively. HSIC-Lasso performs the second-best, with MSE and R^2 values of 5.0412 and 0.8204, respectively. The $\log \mathcal{R}$-L1 method exhibits the weakest performance, with MSE and R^2 values of 5.3042 and 0.8009, correspondingly. It is worth mentioning that these results hold across all aforementioned datasets.

2) Comparison with K-Fold Cross-Validation: Table III presents the performance comparisons of GL_CSE (Best), GL_CSE (Q1), GL_CSE (Q2), and GL_CSE (K-Fold). GL_CSE (Best) represents the checkpoint $\mathcal{C}^{i,j}$ with the best testing performances on MSE and R^2. GL_CSE (Q1) and GL_CSE (Q2) correspond to setting Γ_{KL} and Γ_{F} as the Q1

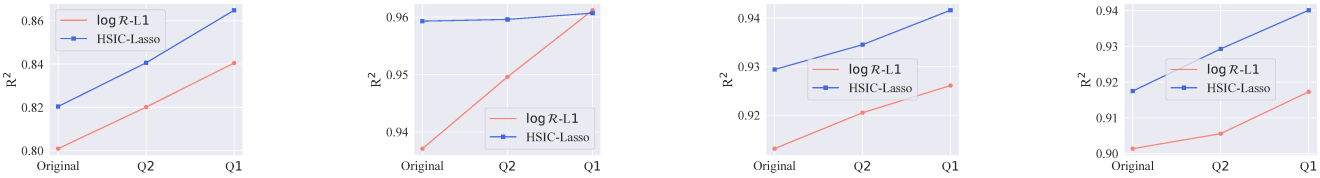
TABLE III: Comparisons on GL_CSE (Best), GL_CSE (Q1), GL_CSE (Q2), and GL_CSE (K-Fold).

		Biological Age	ALL_AML	LUNG	CLL_SUB_111
GL_CSE (Best)	MSE	0.8738	0.0000	0.0271	0.0112
	R^2	0.9894	0.9998	0.9986	0.9990
GL_CSE (Q1)	MSE	0.8799	0.0316	0.1192	0.0486
	R^2	0.9883	0.9991	0.9889	0.9899
GL_CSE (Q2)	MSE	$0.9045^{\downarrow}$	$0.0338^{\downarrow}$	$0.1231^{\downarrow}$	$0.0669^{\downarrow}$
	R^2	$0.9880^{\uparrow}$	$0.9882^{\uparrow}$	$0.9704^{\uparrow}$	$0.9573^{\uparrow}$
GL_CSE (K-Fold)	MSE	1.0612	0.0594	0.1413	0.0694
	R^2	0.9871	0.9669	0.9507	0.9497



(a) Biological Age (b) ALL_AML (c) LUNG (d) CLL_SUB_111

Fig. 1: The MSE on log $\mathcal{R}$-L1, log $\mathcal{R}$-L1 (Q1), log $\mathcal{R}$-L1 (Q2), HSIC-Lasso, HSIC-Lasso (Q1), and HSIC-Lasso (Q2).



(a) Biological Age (b) ALL_AML (c) LUNG (d) CLL_SUB_111

Fig. 2: The R^2 on log $\mathcal{R}$-L1, log $\mathcal{R}$-L1 (Q1), log $\mathcal{R}$-L1 (Q2), HSIC-Lasso, HSIC-Lasso (Q1), and HSIC-Lasso (Q2).

and Q2 of all ϵ_{KL} and ϵ_{F} values, respectively. GL_CSE (K-Fold) means conducting the group Lasso with regular K-fold cross-validation, where $K = 10$. The GL_CSE (K-Fold) algorithm will compute the average of the evaluation measurements over the K validation sets among episodes. The outcomes of our experiments show that GL_CSE (Q1) yields results that are comparable to those of GL_CSE (Best) (indicated in blue). The GL_CSE (Best) indicates the best performance in terms of both metrics in the testing data. Specifically, in the case of the biological age dataset, GL_CSE (Q1) obtains MSE and R^2 of 0.8799 MSE and 0.9883 R^2, respectively, which are comparable to the model with the optimal test performances, i.e., 0.8738 for MSE and 0.9894 for R^2. Additionally, both GL_CSE (Q1) and GL_CSE (Q2) outperform GL_CSE (K-Fold) in terms of the two measurements across all datasets. In particular, for the biological age dataset, GL_CSE (Q2) achieves 0.9045 for MSE and 0.9880 for R^2, which are superior to the regular K-fold cross-validation results, i.e., 1.0612 for MSE and 0.9871 for R^2.

3) Ablation Study:

In this subsection, we will perform an ablation study to demonstrate the importance of the checkpoint selection module in our GL_CSE method. Specifically, we will unravel the importance of the proposed checkpoint selection module using the typical log $\mathcal{R}$-L1 and HSIC-Lasso algorithm. In Figure 1, Standard, Q1, and Q2 represent conducting the regular logistic regression together with our proposed checkpoint selection module. This can be implemented by replacing the group Lasso objective in Line 7 in Algorithm 1 with the regular logistic regression objective. In Figure 2, Standard, Q1, and Q2 represent conducting the HSIC-Lasso regression together with our proposed checkpoint selection module [16]. In Q1 and Q2, Γ_{KL} and Γ_{F} are set to the Q1 and Q2 of all ϵ_{KL} and ϵ_{F} values, respectively. In both Figure 1 and Figure 2, we can see that log $\mathcal{R}$-L1 (Q1) achieves the optimal performance in terms of MSE and R^2, followed by log $\mathcal{R}$-L1 (Q2), followed by the regular log $\mathcal{R}$-L1. For example, in Figure 1a and Figure 2a, log $\mathcal{R}$-L1 (Q2) achieves 5.0604 for MSE and 0.8201 for R^2, which is superior to the regular log $\mathcal{R}$-L1 performance, i.e., 5.3042 for MSE and 0.8009 for R^2, for the biological age dataset. We can also see that log $\mathcal{R}$-L1 (Q1) performs the best, with MSE and R^2 values of 4.8362 and 0.8405, which are better than the values obtained by the regular HSIC-Lasso algorithm, i.e., 5.0412 for MSE and 0.8204 for R^2, respectively (shown in Table II). We also observe that HSIC-Lasso (Q1) achieves the optimal performance in terms of MSE and R^2, followed by HSIC-

Lasso (Q2), and then the regular HSIC-Lasso. For example, in Figure 1a and Figure 2a, HSIC-Lasso (Q2) achieves 4.6714 for MSE and 0.8406 for R^2, which are superior to the regular HSIC-Lasso performance, i.e., 5.0412 for MSE and 0.8204 for R^2, respectively, for the biological age dataset. We can also see that HSIC-Lasso (Q1) achieves the best performance, i.e., 4.0013 for MSE and 0.8649 for R^2, respectively. Thus, this experiment highlights the significance of our proposed checkpoint selection module.

IV. Conclusion

In this paper, we present the GL_CSE algorithm to address two critical issues in HDLSS biological data regression: data sparsity and data drifting. To tackle the data sparsity problem, we employ a group Lasso regression model. To handle the data drifting issue, we propose a checkpoint selection method that extracts essential model checkpoints while training the group Lasso. The checkpoint selection is performed based on two proposed metrics, i.e., the average KL-divergence between training and validation features and the Frobenius error of the covariance matrices between training and validation features. Our experimental results demonstrate that our GL_CSE algorithm outperforms other baseline methods in terms of performance. Furthermore, our proposed checkpoint selection component outperforms traditional K-fold cross-validation. Specifically, on the biological age dataset, GL_CSE (Q2) achieves 0.9045 for MSE and 0.9880 for R^2, respectively, which is superior to the K-fold cross-validation results, i.e., 1.0612 for MSE and 0.9871 for R^2, respectively. Our future work includes (1) developing ensemble models and (2) performing neural architecture search on the selected models set to enhance the regression performance and select important group features simultaneously.

References

[1] Necla Gunduz and Ernest P. Fokoue. Robust classification of high dimension low sample size data. *arXiv: Applications*, 2015.

[2] Khuloud Jaqaman and Gaudenz Danuser. Linking data to models: data regression. *Nature Reviews Molecular Cell Biology*, 7(11):813–819, 2006.

[3] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

[4] Haifeng Li, Tao Jiang, and Keshu Zhang. Efficient and robust feature extraction by maximum margin criterion. *Advances in Neural Information Processing Systems*, 16, 2003.

[5] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys*, 50(6):1–45, 2017.

[6] Juntao Li, Ke Liang, and Xuekun Song. Logistic regression with adaptive sparse group lasso penalty and its application in acute leukemia diagnosis. *Computers in Biology and Medicine*, 141:105154, 2022.

[7] Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[8] Jean-Claude Lorquet. Crossing the dividing surface of transition state theory. iv. dynamical regularity and dimensionality reduction as key features of reactive trajectories. *The Journal of Chemical Physics*, 146(13): 134310, 2017.

[9] Yong Luo, Dacheng Tao, Kotagiri Ramamohanarao, Chao Xu, and Yonggang Wen. Tensor canonical correlation analysis for multi-view dimension reduction. *IEEE Transactions@articlemeier2008group, title=The group lasso for logistic regression, author=Meier, Lukas and Van De Geer, Sara and Bühlmann, Peter, journal=Journal of the Royal Statistical Society Series B: Statistical Methodology, volume=70, number=1, pages=53–71, year=2008, publisher=Oxford University Press on Knowledge and Data Engineering*, 27 (11):3111–3124, 2015.

[10] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993.

[11] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[12] Jens Meiler, Michael Müller, Anita Zeidler, and Felix Schmäschke. Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. *Molecular Modeling Annual*, 7 (9):360–369, 2001.

[13] Rattanawadee Panthong and Anongnart Srivihok. Wrapper feature subset selection for dimension reduction based on ensemble learning algorithm. *Procedia Computer Science*, 72:162–169, 2015.

[14] Liran Shen, Meng Joo Er, and Qingbo Yin. Classification for high-dimension low-sample size data. *Pattern Recognition*, 130:108828, 2022. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2022.108828. URL https://www.sciencedirect.com/science/article/pii/S0031320322003090.

[15] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

[16] Makoto Yamada, Wittawat Jitkrittum, Leonid Sigal, Eric P Xing, and Masashi Sugiyama. High-dimensional feature selection by feature-wise kernelized lasso. *Neural Computation*, 26(1):185–207, 2014.

[17] Hua Yu and Jie Yang. A direct lda algorithm for high-dimensional data—with application to face recognition. *Pattern Recognition*, 34(10):2067–2070, 2001.