Published in final edited form as:
J Comput Graph Stat. 2022 ; 31(4): 1063–1075. doi:10.1080/10618600.2022.2035739.

Variable selection with multiply-imputed datasets: choosing between stacked and grouped methods

Jiacong Du[1], **Jonathan Boss**[1], **Peisong Han**[1], **Lauren J Beesley**[1], **Michael Kleinsasser**[1], **Stephen A Goutman**[2], **Stuart Batterman**[3], **Eva L Feldman**[2], **Bhramar Mukherjee**[1]

[1]Department of Biostatistics, University of Michigan, Ann Arbor, MI

[2]Department of Neurology, University of Michigan, Ann Arbor, MI

[3]Department of Environmental Health Science, University of Michigan, Ann Arbor, MI

Abstract

Penalized regression methods are used in many biomedical applications for variable selection and simultaneous coefficient estimation. However, missing data complicates the implementation of these methods, particularly when missingness is handled using multiple imputation. Applying a variable selection algorithm on each imputed dataset will likely lead to different sets of selected predictors. This paper considers a general class of penalized objective functions which, by construction, force selection of the same variables across imputed datasets. By pooling objective functions across imputations, optimization is then performed jointly over all imputed datasets rather than separately for each dataset. We consider two objective function formulations that exist in the literature, which we will refer to as “stacked” and “grouped” objective functions. Building on existing work, we (a) derive and implement efficient cyclic coordinate descent and majorization-minimization optimization algorithms for continuous and binary outcome data, (b) incorporate adaptive shrinkage penalties, (c) compare these methods through simulation, and (d) develop an R package *miselect.* Simulations demonstrate that the “stacked” approaches are more computationally efficient and have better estimation and selection properties. We apply these methods to data from the University of Michigan ALS Patients Biorepository aiming to identify the association between environmental pollutants and ALS risk. Supplementary materials are available online.

Keywords

Elastic Net; Group LASSO; Majorization-Minimization; Missing Data; Multiple Imputation; Pooled Objective Function

1 Introduction

Variable selection in the presence of missing data is a common problem in applied statistics. Existing statistical methods (Table S1) to deal with both missing data and variable selection fall into two broad classes depending on whether the missing data is

Corresponding author Jiacong Du jiacong@umich.edu.

handled prior to or at the same time as variable selection. While one can, in principle, handle missingness and selection simultaneously (Yang et al., 2005; Garcia et al., 2010; Johnson et al., 2008), in this paper we focus on a different scenario, where missing data have already been handled via multiple imputation and we wish to do variable selection post-imputation. In practice, we often find ourselves in the latter scenario for many reasons. Some examples include: (i) existing data sources or analytical pipelines provide the analyst with already imputed datasets (Schafer, 2001), (ii) data privacy settings where the missingness mechanism depends on sensitive information that is not needed for downstream analyses, and (iii) practical situations where the analyst wants to use an existing imputation software package to generate the multiply imputed datasets to be used across various analyses. That is, in this article we assume that the imputation model is correctly specified, imputed datasets are given to us, and we focus purely on variable selection techniques post-imputation. In particular, the challenge is to guarantee uniform selection of variables across multiply imputed datasets and to generate point estimates for the regression coefficients corresponding to the selected variables.

To address this issue, there are some existing methods in the literature proposed for variable selection post-imputation. The most common approach is to define an ad-hoc rule for harmonizing selection across imputed datasets (Wood et al., 2008; Lachenbruch, 2011; Ghosh et al., 2015). Wood et al. (2008) outlined three rules for finalizing selection after identifying the active set of predictors in each imputed dataset. These rules consider a variable to be selected if it is selected in at least one imputed dataset, if it is selected in all imputed datasets, or if it is selected in at least half of the imputed datasets. Another related approach is to apply variable selection methods on bootstrapped datasets (Heymans et al., 2007; Long and Johnson, 2015; Liu et al., 2016). One such method, proposed by Long and Johnson (2015), imputes missing values in the bootstrapped samples from the original data and then applies penalized regression on each bootstrapped imputed dataset. Proportions of each covariate being selected over all bootstrap imputed datasets are provided and the final active set is determined by thresholding the proportions. The challenge with such thresholding approaches is that there are no clear guidelines on how to choose the threshold. An alternative class of methods are penalized regression estimators whose objective functions involve a joint optimization over all imputed datasets. Wan et al. (2015) proposed a penalized regression estimator which is equivalent to fitting the penalized regression model on the stacked imputed datasets. The stacked method can also be interpreted in terms of maximizing an objective function pooled over imputed datasets, where regression coefficients are assumed to be the same across datasets. Chen and Wang (2013) proposed the multiple imputation-least absolute shrinkage and selection operator (MI-LASSO) which utilizes a group LASSO penalty to group regression coefficients of the same variable across imputed datasets together, thus arriving at uniform variable selection across imputed datasets. This grouped approach is less restrictive than the stacked approach, in that the parameter values are allowed to differ across imputed datasets. The stacked and grouped approaches are appealing because they handle variable selection and regression coefficient estimation simultaneously, and no ad hoc rules are required to determine the final active set.

Despite the advantages of the stacked and grouped approaches, automated implementation of these two methods (Wan et al., 2015; Chen and Wang, 2013) has been limited in practice. There is a lack of easy-to-use software for both approaches, and for variable selection with multiple imputed datasets in general. More specifically, off-the-shelf R packages such as *glmnet* (Hastie and Qian, 2014) and *gcdnet* (Yang et al., 2017) are inadequate for optimizing the stacked objective function proposed in Wan et al. (2015). Namely, the R package *glmnet* does not exclusively apply adaptive weights to the L_1 penalty, as in adaptive elastic net, and the *gcdnet* package does not implement observation weights. As for the grouped approach with the group LASSO penalty function, *gglasso* (Yang and Zou, 2015) cannot be directly used without changing the data structure. The dimension of the new design matrix scales linearly with the number of imputed datasets and introduces many zero entries, which is computationally inefficient. Furthermore, existing approaches are targeted towards continuous outcomes and have not been extended to discrete/categorical outcomes that are abundant in practice. Even for continuous outcomes, the algorithm for MI-LASSO in Chen and Wang (2013) has an important limitation. Specifically, the existing MI-LASSO implementation is an optimization procedure that relies on a local quadratic approximation to the L_1 penalty, which converts the penalized objective function into multiple ridge regressions. Coefficient estimates are always non-zero, requiring a manual threshold for setting small coefficient estimates exactly to zero.

To address the aforementioned limitations, we extend existing work on the stacked and grouped approaches for handling variable selection with multiple imputed datasets to incorporate binary outcomes and adaptive penalties (Section 2). We derive cyclic coordinate descent algorithms for optimizing the pooled objective functions from the stacked approach and majorization-minimization (MM) algorithms coupled with block coordinate descent updates to obtain optimizers of the pooled objective functions from the grouped approach (Section 3). Unlike the optimization routine in Chen and Wang (2013), our proposed methods provide exact shrinkage to zero without any ad hoc thresholding. We provide an R package *miselect* available on the Comprehensive R Archive Network (CRAN), allowing users to easily implement the proposed methods. In our motivating example, we apply these methods to identify persistent organic pollutants (POPs) associated with Amyotrophic Lateral Sclerosis (ALS) susceptibility using data collected from the University of Michigan ALS Patients Biorepository (Section 4). Seventeen of the 23 POPs have between 10% to 60% below their respective detection limits, making complete-case analysis infeasible. Finally, through a simulation study, we compare the performance of the proposed methods in terms of variable selection and estimation accuracy (Section 5).

2 Methods

We will first review and illustrate the issues with identifying an active set of predictors when fitting separate penalized regressions on each individual imputed dataset. We will then present the stacked and grouped approaches for uniform variable selection with imputed datasets. Let $\boldsymbol{X}_d$ denote the $n \times p$ matrix of predictor variables and $\boldsymbol{Y}_d$ be the $n \times 1$ vector of responses for the d-th imputed dataset ($d = 1, \ldots D$). Let $\boldsymbol{X}_{d,i}$ indicate the $p \times 1$ covariate vector for the i-th observation in the d-th imputed dataset, $Y_{d,i}$ be the response for the

i-th observation in the d-th imputed dataset, and $X_{d,ij}$, denote the j-th covariate for the i-th observation in the d-th imputed dataset.

2.1 Penalized regression on individual datasets

A common approach in practice is to use ad hoc rules for determining the final set of selected variables. Often this proceeds by fitting a penalized regression procedure on each imputed dataset separately:

$$\hat{\boldsymbol{\theta}}_d = \underset{\boldsymbol{\theta}_d}{\operatorname{argmin}}\left\{-\frac{1}{n}\sum_{i=1}^{n}\log L\left(\boldsymbol{\theta}_d \mid Y_{d,i}, \boldsymbol{X}_{d,i}\right) + \lambda \boldsymbol{P}_\alpha(\boldsymbol{\beta}_d)\right\} \qquad (1)$$

for $d = 1, \ldots, D$ where log L is the log-likelihood function, $\boldsymbol{\beta}_d$ is the p × 1 vector of regression coefficients for the d-th dataset, and $\boldsymbol{\theta}_d = (\mu_d, \boldsymbol{\beta}_d)$ is the vector of regression parameters including the intercept and coefficients for the d-th imputed dataset. $P_\alpha(\boldsymbol{\beta}_d)$ is the penalty function parameterized by α and $\lambda \in (0, \infty)$ which are tuning parameters that control the relative contribution of the penalty to the overall optimization problem.

In this paper, we will focus on four penalty functions: LASSO (Tibshirani, 1996), adaptive LASSO (aLASSO) (Zou, 2006), elastic net (ENET) (Zou and Hastie, 2005) and adaptive elastic net (aENET) (Zou and Zhang, 2009). The penalty functions can be expressed as:

1. LASSO: $P_\alpha(\boldsymbol{\beta}_d) = \sum_{j=1}^{p}\left|\beta_{d,j}\right|$
2. aLASSO : $P_\alpha(\boldsymbol{\beta}_d) = \sum_{j=1}^{p}\hat{a}_{d,j}\left|\beta_{d,j}\right|$
3. ENET: $P_\alpha(\boldsymbol{\beta}_d) = \alpha\sum_{j=1}^{p}\left|\beta_{d,j}\right| + (1-\alpha)\sum_{j=1}^{p}\beta_{d,j}^2$
4. aENET: $P_\alpha(\boldsymbol{\beta}_d) = \alpha\sum_{j=1}^{p}\hat{a}_{d,j}\left|\beta_{d,j}\right| + (1-\alpha)\sum_{j=1}^{p}\beta_{d,j}^2$

$\beta_{d,j}$ is the coefficient for the j-th covariate in the d-th dataset. The intuitive appeal of adaptive weights is that they allow differential penalization of each covariate based on an initial estimate of the regression coefficient vector $\hat{\boldsymbol{\beta}}_d^0$ (smaller $\hat{\beta}_{d,j}^0$ implies a harsher penalty on $\beta_{d,j}$). Moreover, adaptive penalties are known to address estimation and selection consistency issues that have been observed for non-adaptive penalties (Zou, 2006; Zou and Zhang, 2009). Here, the adaptive weight $\hat{a}_{d,j} = \left(\left|\hat{\beta}_{d,j}^0\right| + 1/n\right)^{-\gamma}$ for some $\gamma > 0$, where $\hat{\beta}_{d,j}^0$ is an initial estimate of $\beta_{d,j}$, is often determined from ordinary least squares (OLS) or maximum likelihood estimation when p is smaller than n. If p is larger than n, $\hat{\beta}_{d,j}^0$ can be obtained using LASSO or ENET depending on the correlation structure of the predictors. For the purposes of this paper, we use ENET initial values to calculate the adaptive weights for both aLASSO and aENET. To avoid tuning on γ, we follow Zou and Zhang (2009), which fixes $\gamma = \lceil 2v/1 - v\rceil + 1$ where $\upsilon = \log(p)/\log(n)$. The $1/n$ term prevents division by zero for initial regression coefficient estimates that are exactly equal to zero.

For illustrative purposes, suppose that there are three imputed datasets and that LASSO is fit separately on each imputed dataset. Furthermore, assume that $\boldsymbol{X}_d$ is an n × 2 matrix for all

$d = 1, 2, 3$, i.e., we are applying LASSO shrinkage to coefficients of only two covariates. Figure 1 visualizes the geometry of the constrained region in this hypothetical scenario. As shown in the figure, the maximum likelihood estimates for each of the imputed datasets are slightly different and, when shrunk to the constrained region, lead to two cases: (i) $\hat{\beta}_{d,1} = 0$ and $\hat{\beta}_{d,2} \neq 0$ and (ii) $\hat{\beta}_{d,1} \neq 0$ and $\hat{\beta}_{d,2} = 0$. This toy example illustrates the fundamental problem of trying to harmonize variable selection across imputed datasets without borrowing information across imputed datasets.

2.2 Stacked objective functions

In a stacked objective function, we sum the loss functions for each of the imputed datasets together and jointly optimize the collective objective function:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\left\{-\frac{1}{n}\sum_{d=1}^{D}\sum_{i=1}^{n}\log L\left(\boldsymbol{\theta} \mid Y_{d,i}, \boldsymbol{X}_{d,i}\right) + \lambda P_{\alpha}(\boldsymbol{\beta})\right\} \quad (2)$$

Note that $\boldsymbol{\theta}$ is not indexed by d. This implies that optimizing the pooled objective function will result in one estimated parameter vector $\hat{\boldsymbol{\theta}}$, thereby enforcing uniform selection across all imputed datasets.

A nice feature of the stacked objective function is that optimization is straightforward. Namely, optimization of the stacked objective function is equivalent to stacking the imputed datasets and fitting the desired penalized regression algorithm on the stacked imputed datasets with existing software. Therefore, the stacked objective function provides a framework for pooling penalized regression estimates across imputed datasets for a general class of objective functions. However, stacking all imputed datasets can be viewed as artificially increasing the sample size. A common way to address this is to add an observation weight, $o_i = 1/D$, so that the total weight for each subject in the stacked dataset sums up to one. An alternative observational weight specification, proposed in Wan et al. (2015), is $o_i = f_i/D$, where f_i is the number of observed predictors out of the total number of predictors for subject i, which accounts for varying degrees of missing information for each subject. That being said, upweighting subjects with less missingness and downweighting subjects with more missingness can, in some sense, be viewed as making the optimization more like complete-case analysis, which might be problematic for Missing at Random (MAR) and Missing not at Random (MNAR) scenarios. Going forward, we consider both equal weights ($o_i = 1/D$) and the observation weights proposed in Wan et al. (2015) ($o_i = f_i/D$). The weighted stacked objective function can be written as:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\left\{-\frac{1}{n}\sum_{d=1}^{D}\sum_{i=1}^{n}o_i\log L\left(\boldsymbol{\theta} \mid Y_{d,i}, \boldsymbol{X}_{d,i}\right) + \lambda P_{\alpha}(\boldsymbol{\beta})\right\} \quad (3)$$

In (3), we directly sum the log-likelihoods ignoring the correlation for the same observation across all imputed datasets. This is not a problem for point estimation. Averaging over the number of imputed datasets D can be viewed as obtaining the expected log-likelihood conditional on the observed data and then subsequently averaging over all *n* observations leads to the conditional log-likelihood given the observed data. This strategy of defining an objective function summed across correlated imputed datasets is well-founded in the statistical literature on missing data (Wang and Robins, 1998; Robins and Wang, 2000). Ignoring the dependence between these datasets will have an impact on estimation of standard errors, but the coefficient estimator will still be consistent (Wang and Robins, 1998; Robins and Wang, 2000).

From this stacked approach, we extend the penalty functions to LASSO, aLASSO, ENET and aENET. When equal weights are used, the names for different versions of the stacked methods are as follows:

1. Stacked LASSO (SLASSO): $P_\alpha(\boldsymbol{\beta}) = \sum_{j=1}^{p} |\beta_j|$
2. Stacked adaptive LASSO (SaLASSO): $P_\alpha(\boldsymbol{\beta}) = \sum_{j=1}^{p} \hat{a}_j |\beta_j|$
3. Stacked elastic net (SENET): $P_\alpha(\boldsymbol{\beta}) = \alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2$
4. Stacked adaptive elastic net (SaENET): $P_\alpha(\boldsymbol{\beta}) = \alpha \sum_{j=1}^{p} \hat{a}_j |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2$

where $\hat{a}_j = \left(\left|\hat{\beta}_j\right| + 1/(nD)\right)^{-\gamma}$ and $\hat{\beta}_j$ is estimated through SENET. Following Zou and Zhang (2009), γ is fixed to be $\lceil 2\upsilon/1 - \upsilon \rceil + 1$, where $v = \log(p)/\log(nD)$.

When $o_i = f_i/D$, the penalized methods are named SLASSO(w), SaLASSO(w), SENET(w) and SaENET(w), respectively. Adaptive weights for the stacked approach with the observation weights are calculated using SENET(w).

2.3 Grouped objective functions

An alternative to the stacked objective function is the grouped objective function. This method imposes uniform variable selection across imputed datasets by adding a group LASSO penalty across imputed datasets (Chen and Wang, 2013; Geronimi and Saporta, 2017; Yuan and Lin, 2006). The optimizer of the grouped objective function can be mathematically expressed as:

$$\left(\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_D\right) = \underset{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_D}{\operatorname{argmin}} \left\{ -\frac{1}{n} \sum_{d=1}^{D} \sum_{i=1}^{n} \log L\left(\boldsymbol{\theta}_d \mid Y_{d,i}, \boldsymbol{X}_{d,i}\right) + \lambda P(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_D) \right\} \quad (4)$$

where λ ∈ (0, ∞) is a tuning parameter. Chen et al. (2013) originally formulated a special case of the grouped objective function known as MI-LASSO, where the penalty function is:

$$P(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_D) = \sum_{j=1}^{p} \sqrt{\sum_{d=1}^{D} \beta_{d,j}^2}$$

In the grouped objective function, the parameter vector is now indexed by *d*, meaning that $\boldsymbol{\theta}_1 \neq \ldots \neq \boldsymbol{\theta}_D$. Although the $\boldsymbol{\theta}_d$'s are not identical, for any fixed *j*, the group LASSO penalty jointly shrink all $\beta_{d,j}$'s to zero, i.e. $\beta_{1,j} = \ldots = \beta_{D,j} = 0$. This allows for uniform selection across imputed datasets but also allows for variability in the non-zero estimated coefficients across imputed datasets. Based on Chen et al. (2013), we consider the following penalties:

1. Group LASSO (GLASSO): $P(\beta_1, \beta_2, \ldots, \beta_D) = \sum_{j=1}^{p} \sqrt{\sum_{d=1}^{D} \beta_{d,j}^2}$
2. Group adaptive LASSO (GaLASSO): $P(\beta_1, \beta_2, \ldots, \beta_D) = \sum_{j=1}^{p} \hat{a}_j \sqrt{\sum_{d=1}^{D} \beta_{d,j}^2}$

where $\hat{a}_j = \left(\sqrt{\sum_{d=1}^{D} \hat{\beta}_{d,j}^2} + 1/(nD)\right)^{-\gamma} \cdot \hat{\beta}_{d,j}$ is estimated from GLASSO, $\gamma = \lceil 2v/1 - v \rceil + 1$, and $v = \log(pD)/\log(nD)$ (Zou and Zhang, 2009).

Remark: We do not extend the grouped LASSO based approaches to grouped ENET based approaches because grouping itself provides constraints similar to the ENET penalty. How to effectively account for correlations among predictors using the grouped penalty remains an open question.

3 Optimization

In this section, we outline the optimization routines for SaENET(w) and GaLASSO with binary outcomes. To optimize the SaENET(w) objective function, we use local quadratic approximation coupled with a cyclic coordinate descent algorithm (Friedman et al., 2010). To obtain an optimizer of the GaLASSO objective function, we use a MM algorithm combined with block coordinate descent updates to handle the group LASSO component of the penalty function. The other objective functions listed in Section 2 are special cases of SaENET(w) and GaLASSO, and therefore can be optimized with minor simplifications (see Figure 2). Namely, when $\alpha = 1$, SaENET(w) reduces to SaLASSO(w), and SENET(w) reduces to SLASSO(w). Furthermore, when $\hat{a}_j = 1$ for $j = 1, 2, .., p$, SaENET(w) reduces to SENET(w) and GaLASSO reduces to GLASSO. Optimizing the stacked objective function with equal weights can be achieved by setting $o_i = 1/D$.

3.1 Optimization of SaENET(w)

Without loss of generality, suppose that the variables in the design matrix are standardized after stacking all the imputed datasets. That is, $\sum_{d=1}^{D} \sum_{i=1}^{n} x_{d,ij} = 0$, $\frac{1}{n} \sum_{d=1}^{D} \sum_{i=1}^{n} x_{d,ij}^2 = 1$, for $j = 1, 2, \ldots, p$. Let $\boldsymbol{\eta} = \left(\boldsymbol{\eta}_1^T, \boldsymbol{\eta}_2^T, \ldots, \boldsymbol{\eta}_D^T\right)^T$ be the linear predictor based on $\boldsymbol{\theta} = \left(\mu, \boldsymbol{\beta}^T\right)^T$ such that $\eta_{d,i} = \mu + x_{d,i}^T \boldsymbol{\beta}$. Then the sum of the weighted loss functions can be expressed as:

$$L(\boldsymbol{\eta}) = \frac{1}{n} \sum_{d=1}^{D} \sum_{i=1}^{n} o_i \left\{-y_{d,i} \eta_{d,i} + \log\left(1 + \exp\left(\eta_{d,i}\right)\right)\right\}$$

Let $\boldsymbol{\eta}^{(t)}$ denote the linear predictor at the t-th iteration. Following Friedman et al. (2010), we use a Taylor expansion at $\boldsymbol{\eta}^{(t)}$ to construct a quadratic approximation to the loss function:

$$L_Q\left(\boldsymbol{\eta} \mid \boldsymbol{\eta}^{(t)}\right) = \frac{1}{2n}\sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i}\left(\tilde{y}_{d,i} - \mu - x_{d,i}^T\boldsymbol{\beta}\right)^2 + C\left(\boldsymbol{\eta}^{(t)}\right)^2$$

where

$$\tilde{y}_{d,i} = \mu^{(t)} + x_{d,i}^T\boldsymbol{\beta}^{(t)} + \frac{y_{d,i} - \tilde{p}(x_{d,i})}{\tilde{p}(x_{d,i})(1 - \tilde{p}(x_{d,i}))} \tag{5}$$

is the working response, $w_{d,i} = \tilde{p}(x_{d,i})(1 - \tilde{p}(x_{d,i}))$ is a subject weight that is specific to each imputed dataset, and

$$\tilde{p}(x_{d,i}) = P(Y_{d,i} = 1 \mid X_{d,i} = x_{d,i}) = \left(1 + \exp\left(-\left(\mu^{(t)} + x_{d,i}^T\boldsymbol{\beta}^{(t)}\right)\right)\right)^{-1}.$$

Going forward we will use, $O_Q = L_Q\left(\boldsymbol{\eta} \mid \boldsymbol{\eta}^{(t)}\right) + \lambda P_\alpha(\boldsymbol{\beta})$, as shorthand notation for the objective function after quadratic approximation. We then use coordinate descent to solve:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, O_Q = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\left\{L_Q\left(\boldsymbol{\eta} \mid \boldsymbol{\eta}^{(t)}\right) + \lambda P_{\boldsymbol{\alpha}}(\boldsymbol{\beta})\right\}$$

The derivative of the approximate objective function with respect to the intercept parameter is:

$$\frac{\partial O_Q}{\partial \mu} = -\frac{1}{n}\sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i}\left(\tilde{y}_{d,i} - \mu - x_{d,i}^T\boldsymbol{\beta}\right) = -\frac{1}{n}\sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i}\left(\tilde{y}_{d,i} - x_{d,i}^T\boldsymbol{\beta}\right) + \frac{1}{n}\sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i}\mu$$

Let $z_0 = \sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i}\left(\tilde{y}_{d,i} - x_{d,i}^T\boldsymbol{\beta}\right)$. Then μ can be updated as:

$$\mu^{(t+1)} \leftarrow \frac{z_0}{\sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i}} \tag{6}$$

If $\beta_j > 0$,

$$\frac{\partial O_Q}{\partial \beta_j} = -\frac{1}{n}\sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i} x_{d,ij}\left(\tilde{y}_{d,i} - \mu - x_{d,i(-j)}^T\boldsymbol{\beta}_{(-j)}\right) + \frac{1}{n}\sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i} x_{d,ij}^2\beta_j + \lambda\alpha\hat{a}_j + 2\lambda(1 - \alpha)\beta_j$$

where $x_{d,i(-j)}$ refers to the value of the covariate vector for the *i*-th observation in the *d*-th imputed dataset after removing the *j*-th covariate, and $\boldsymbol{\beta}_{(-j)}$ refers to the regression coefficient vector $\boldsymbol{\beta}$ without the *j*-th entry. If $\beta_j < 0$ then the derivative of O_Q with respect to β_j is the same as the derivative when $\beta_j > 0$, the only exception being that the $\lambda\alpha\hat{a}_j$ term becomes $-\lambda\alpha\hat{a}_j$. Setting $z_j = \sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i} x_{d,ij}\left(\tilde{y}_{d,i} - \mu - x_{d,i(-j)}^T \boldsymbol{\beta}_{(-j)}\right)$, then β_j can be updated as:

$$\beta_j^{(t+1)} \leftarrow \frac{S\left(\frac{1}{n}z_j, \lambda\alpha\hat{a}_j\right)}{\frac{1}{n}\sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i} x_{d,ij}^2 + 2\lambda(1-\alpha)} \tag{7}$$

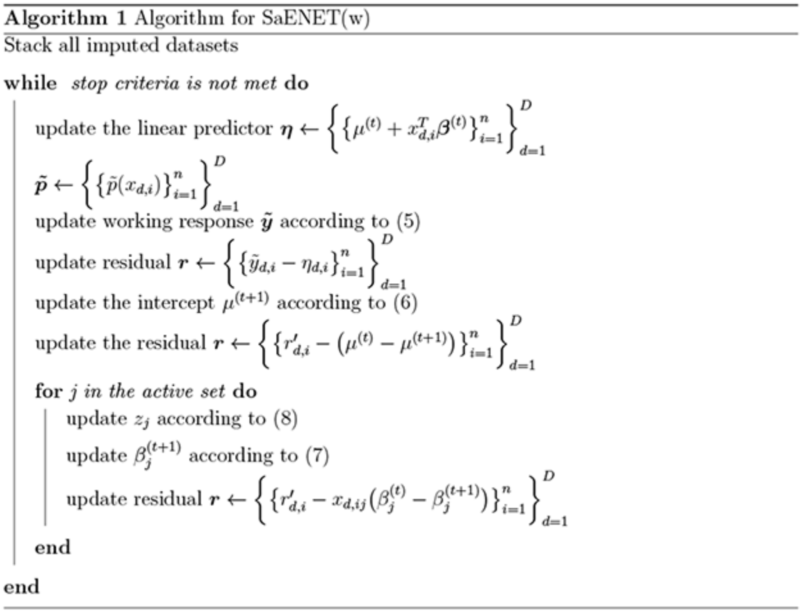
where $S(z, \lambda)$ is the soft-thresholding operator:

$$S(z,\lambda) = \begin{cases} 0 & \text{if} \, |z| \leq \lambda \\ z - \lambda & \text{if} \, z > \lambda \\ z + \lambda & \text{if} \, z < -\lambda \end{cases}$$

A summary of the optimization routine for SaENET(w) is presented in Algorithm 1. One thing to note is that we never directly compute z_j after each update of β_j. A more computationally efficient approach is to update z_j through the residual $r_{d,i} = \tilde{y}_{d,i} - \mu - x_{d,i}^T \boldsymbol{\beta}$. Specifically,

$$\begin{gathered} z_j \leftarrow \sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i} x_{d,ij} r_{d,i} + \sum_{d=1}^{D}\sum_{i=1}^{n} o_i w_{d,i} x_{d,ij}^2 \beta_j^{(t)} \\ r_{d,i} \leftarrow r'_{d,i} - x_{d,ij}\left(\beta_j^{(t)} - \beta_j^{(t+1)}\right) \end{gathered} \tag{8}$$

where $r'_{d,i}$ indicates the previous estimate of $r_{d,i}$ (we do not use the (t) superscript notation here because $r_{d,i}$ is updated multiple times within the same iteration). Updating the intercept parameter μ is similar, in that we assign $r_{d,i} \leftarrow r'_{d,i} - \left(\mu^{(t)} - \mu^{(t+1)}\right)$ and update z_0 accordingly. Once a coefficient is shrunk to zero, it will stay at zero for the remaining iterations.



```
Algorithm 1 Algorithm for SaENET(w)
Stack all imputed datasets
while stop criteria is not met do
    update the linear predictor $\boldsymbol{\eta} \leftarrow \left\{\{\mu^{(t)} + x_{d,i}^T \boldsymbol{\beta}^{(t)}\}_{i=1}^n\right\}_{d=1}^D$
    $\tilde{\boldsymbol{p}} \leftarrow \left\{\{\tilde{p}(x_{d,i})\}_{i=1}^n\right\}_{d=1}^D$
    update working response $\tilde{\boldsymbol{y}}$ according to (5)
    update residual $\boldsymbol{r} \leftarrow \left\{\{\tilde{y}_{d,i} - \eta_{d,i}\}_{i=1}^n\right\}_{d=1}^D$
    update the intercept $\mu^{(t+1)}$ according to (6)
    update the residual $\boldsymbol{r} \leftarrow \left\{\{r'_{d,i} - (\mu^{(t)} - \mu^{(t+1)})\}_{i=1}^n\right\}_{d=1}^D$
    for $j$ in the active set do
        update $z_j$ according to (8)
        update $\beta_j^{(t+1)}$ according to (7)
        update residual $\boldsymbol{r} \leftarrow \left\{\{r'_{d,i} - x_{d,ij}(\beta_j^{(t)} - \beta_j^{(t+1)})\}_{i=1}^n\right\}_{d=1}^D$
    end
end
```

3.2 Optimization of GaLASSO

Although Chen and Wang (2013) initially proposed the idea of grouped objective functions, their optimization procedure relies on a local quadratic approximation argument to rewrite the grouped objective function as the sum of D separate ridge regressions. Because the ridge penalty does not shrink regression coefficient estimates all the way to zero, the user is then forced to threshold the resulting regression coefficient estimates in an ad hoc manner. Another limitation of the algorithm proposed in Chen and Wang (2013) is that they only consider continuous outcome variables. In many biomedical applications, including our motivating example, the outcome is a binary indicator of disease status. To address both of these concerns, we developed a MM algorithm for GaLASSO with binary outcome data. The primary computational advantage of the MM-Algorithm is that it allows us to transform the optimization of a non-linear loss function into an optimization of a linear function, called the majorizing function. Block coordinate descent can then be applied to optimize the linear majorizing function, where the blocks correspond to regression coefficient estimates for each covariate across all imputed datasets, i.e., $(\beta_{1,j}, \beta_{2,j}, \ldots, \beta_{D,j})$ for the j-th covariate.

Without loss of generality, we standardize the data by letting $\sum_{i=1}^n x_{d,ij} = 0, n^{-1}\sum_{i=1}^n x_{d,ij}^2 = 1$, for $j = 1, 2, \ldots, p$ and $d = 1, 2, \ldots, D$. Let $\boldsymbol{\eta} = \left(\boldsymbol{\eta}_1^T, \boldsymbol{\eta}_2^T, \ldots, \boldsymbol{\eta}_D^T\right)^T$ be the linear predictor based on $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_D)$, where $\boldsymbol{\theta}_d = \left(\mu_d, \boldsymbol{\beta}_d^T\right)^T$, $\boldsymbol{\eta}_d = (\eta_{d,1}, \eta_{d,2}, \ldots, \eta_{d,n})^T$ and $\eta_{d,i} = \mu_d + \beta_{d,1}x_{d,i1} + \ldots + \beta_{d,p}x_{d,ip}$. For binary outcome data, the sum of loss functions in (4) can be written as:

$$L(\boldsymbol{\eta}) = \sum_{d=1}^{D}\sum_{i=1}^{n}\{-y_{d,i}\eta_{d,i} + \log(1 + \exp(\eta_{d,i}))\}$$

Let $\boldsymbol{\eta}^{(t)}$ be the linear predictor at the t-th iteration, $L(\boldsymbol{\eta}^{(t)})$ be the loss function given $\boldsymbol{\eta}^{(t)}$, and $\tilde{L}\left(\boldsymbol{\eta} \mid \boldsymbol{\eta}^{(t)}\right)$ be the majorizing function of $L(\boldsymbol{\eta}^{(t)})$ (Breheny and Huang, 2015), which can be expressed as:

$$\tilde{L}\left(\boldsymbol{\eta} \mid \boldsymbol{\eta}^{(t)}\right) = L\left(\boldsymbol{\eta}^{(t)}\right) + \left(\boldsymbol{\eta} - \boldsymbol{\eta}^{(t)}\right)^{T} \nabla L\left(\boldsymbol{\eta}^{(t)}\right) + \frac{v}{2}\left(\boldsymbol{\eta} - \boldsymbol{\eta}^{(t)}\right)^{T}\left(\boldsymbol{\eta} - \boldsymbol{\eta}^{(t)}\right).$$

Here, $v = max_d max_i sup_\eta \left\{\nabla^2 L_{d,i}(\eta)\right\} = 0.25$ (Breheny and Huang, 2015). Ignoring constants not involving $\boldsymbol{\eta}$, $\tilde{L}\left(\boldsymbol{\eta} \mid \boldsymbol{\eta}^{(t)}\right)$ can be written in terms of $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_D)$:

$$\tilde{L}\left(\boldsymbol{\eta} \mid \boldsymbol{\eta}^{(t)}\right) = \tilde{L}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_D) \propto \frac{v}{2n} \sum_{d=1}^{D} \sum_{i=1}^{n} \left(\tilde{y}_{d,i} - \mu_d - x_{d,i}^{T} \boldsymbol{\beta}_d\right)^2$$

where

$$\tilde{y}_{d,i} = \mu_d^{(t)} + x_{d,i}^{T} \boldsymbol{\beta}_d^{(t)} + \frac{y_{d,i} - \tilde{p}\left(x_{d,i}\right)}{v} \tag{9}$$

is the working response and

$$\tilde{p}\left(x_{d,i}\right) = P\left(Y_{d,i} = 1 \mid X_{d,i} = x_{d,i}\right) = \left(1 + \exp\left(-\left(\mu_d^{(t)} + x_{d,i}^{T} \boldsymbol{\beta}_d^{(t)}\right)\right)\right)^{-1}.$$

Going forward, let $M_Q = \tilde{L}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_D) + \lambda P(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_D)$ denote the penalized majorizing function, which we want to optimize:

$$\underset{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_D}{\operatorname{argmin}} M_Q = \underset{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_D}{\operatorname{argmin}} \left\{\tilde{L}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_D) + \lambda P(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_D)\right\}$$

To derive the block coordinate descent updates for the majorizing function optimization, we first need to introduce some new notation to distinguish coefficients within an imputed dataset from those across the imputed datasets. Let $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_D)^T$ be the vector of all intercept parameters across the imputed datasets, let $\boldsymbol{\beta}_{\cdot,j} = (\beta_{1,j}, \beta_{2,j}, \ldots, \beta_{D,j})^T$, for $j = 1, 2, \ldots, p$ denote the vector of the j-th regression coefficients across all imputed datasets, and let $\boldsymbol{\beta}_{d,\cdot} = (\beta_{d,1}, \beta_{d,2}, \ldots, \beta_{d,p})^T$ for $d = 1, 2, \ldots, D$ indicate the regression coefficients for the d-th imputed dataset. If we take the subdifferential of M_Q with respect to $\boldsymbol{\mu}$, we get:

$$\frac{\partial M_Q}{\partial \boldsymbol{\mu}} = -\frac{v}{n} \boldsymbol{z}_0 + v \boldsymbol{\mu}$$

where

$$z_0 = \begin{bmatrix} \sum_{i=1}^{n} \left(\tilde{y}_{1,i} - x_{1,i}^T \beta_{1,\cdot}\right) \\ \sum_{i=1}^{n} \left(\tilde{y}_{2,i} - x_{2,i}^T \beta_{2,\cdot}\right) \\ \vdots \\ \sum_{i=1}^{n} \left(\tilde{y}_{D,i} - x_{D,i}^T \beta_{D,\cdot}\right) \end{bmatrix}$$

Since $x_{d,ij}$, have been centered, the intercept can be updated as:

$$\boldsymbol{\mu}^{(t+1)} \leftarrow \begin{bmatrix} \frac{1}{n} \sum_{i=1}^{n} \tilde{y}_{1,i} \\ \frac{1}{n} \sum_{i=1}^{n} \tilde{y}_{2,i} \\ \vdots \\ \frac{1}{n} \sum_{i=1}^{n} \tilde{y}_{D,i} \end{bmatrix} \tag{10}$$

We now derive the update for the block of coefficients $\boldsymbol{\beta}_{\cdot,j}$, $j = 1, \ldots, p$. If $\boldsymbol{\beta}_{\cdot,j} \neq \mathbf{0}$, the subdifferential of M_Q with respect to $\boldsymbol{\beta}_{\cdot,j}$ is:

$$\frac{\partial M_Q}{\partial \boldsymbol{\beta}_{\cdot,j}} = -\frac{v}{n} z_j + v \boldsymbol{\beta}_{\cdot,j} + \lambda \hat{a}_j \frac{\boldsymbol{\beta}_{\cdot,j}}{\|\boldsymbol{\beta}_{\cdot,j}\|}$$

where

$$z_j = \begin{bmatrix} \sum_{i=1}^{n} x_{1,ij}\left(\tilde{y}_{1,i} - \mu_1 - x_{1,i(-j)}^T \beta_{1,(-j)}\right) \\ \sum_{i=1}^{n} x_{2,ij}\left(\tilde{y}_{2,i} - \mu_2 - x_{2,i(-j)}^T \beta_{2,(-j)}\right) \\ \vdots \\ \sum_{i=1}^{n} x_{D,ij}\left(\tilde{y}_{D,i} - \mu_D - x_{D,i(-j)}^T \beta_{D,(-j)}\right) \end{bmatrix}$$

Here, $x_{d,i(-j)}$ is the vector of covariates for the i-th observation in the d-th dataset after removing $x_{d,ij}$, and $\boldsymbol{\beta}_{d,(-j)}$ is the corresponding vector of regression coefficients. The form of the partial derivative with respect to $\boldsymbol{\beta}_{\cdot,j}$ indicates that the $\boldsymbol{\beta}_{\cdot,j}$ update must lie on the line segment joining the zero vector, $\mathbf{0}$, and $\mathbf{z}_j$. Therefore, $\boldsymbol{\beta}_{\cdot,j}$ can be updated as:

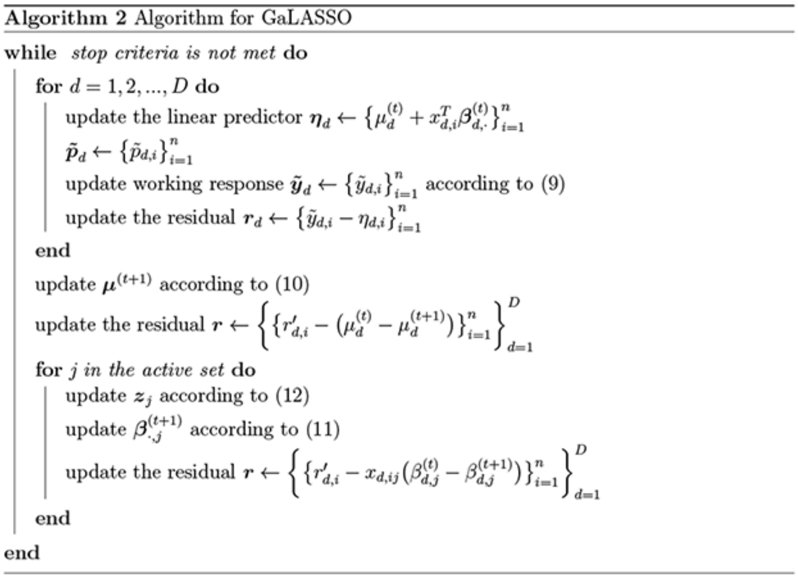
$$\boldsymbol{\beta}_{\cdot,j}^{(t+1)} \leftarrow \frac{1}{v} S\left(\frac{v}{n}\left\|\mathbf{z}_j\right\|, \lambda \hat{a}_j\right) \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|} \tag{11}$$

As with SaENET(w), we do not directly calculate $\mathbf{z}_j$ but use the residual $r_{d,i}$ to update $\mathbf{z}_j$. Let $r_{d,i} = \tilde{y}_{d,i} - \mu_d - x_{d,i}^T \beta_{d,\cdot}$. Then

$$\boldsymbol{z}_j \leftarrow \begin{bmatrix} z_{1,j} \\ z_{2,j} \\ \vdots \\ z_{D,j} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_{1,ij} r_{1,i} + n\beta_{1,j}^{(t)} \\ \sum_{i=1}^{n} x_{2,ij} r_{2,i} + n\beta_{2,j}^{(t)} \\ \vdots \\ \sum_{i=1}^{n} x_{D,ij} r_{D,i} + n\beta_{D,j}^{(t)} \end{bmatrix} \tag{12}$$

$$r_{d,i} \leftarrow r'_{d,i} - x_{d,ij}\left(\beta_{d,j}^{(t)} - \beta_{d,j}^{(t+1)}\right)$$

Updating the intercept parameter $\boldsymbol{\mu}$ is similar, in that we assign $r_{d,i} \leftarrow r'_{d,i} - \left(\mu_d^{(t)} - \mu_d^{(t+1)}\right)$ and update z_0 accordingly. Once the block of regression coefficients corresponding to the *j*-th covariate is shrunk to zero, it will stay at zero for the remaining iterations. A summary of the optimization routine is described in Algorithm 2.



Algorithm 2 Algorithm for GaLASSO

```
while stop criteria is not met do
    for $d = 1, 2, ..., D$ do
        update the linear predictor $\boldsymbol{\eta}_d \leftarrow \{\mu_d^{(t)} + x_{d,i\cdot}^T \boldsymbol{\beta}_{d,\cdot}^{(t)}\}_{i=1}^n$
        $\tilde{\boldsymbol{p}}_d \leftarrow \{\tilde{p}_{d,i}\}_{i=1}^n$
        update working response $\tilde{\boldsymbol{y}}_d \leftarrow \{\tilde{y}_{d,i}\}_{i=1}^n$ according to (9)
        update the residual $\boldsymbol{r}_d \leftarrow \{\tilde{y}_{d,i} - \eta_{d,i}\}_{i=1}^n$
    end
    update $\boldsymbol{\mu}^{(t+1)}$ according to (10)
    update the residual $\boldsymbol{r} \leftarrow \left\{\{r'_{d,i} - (\mu_d^{(t)} - \mu_d^{(t+1)})\}_{i=1}^n\right\}_{d=1}^D$
    for $j$ in the active set do
        update $\boldsymbol{z}_j$ according to (12)
        update $\boldsymbol{\beta}_{\cdot,j}^{(t+1)}$ according to (11)
        update the residual $\boldsymbol{r} \leftarrow \left\{\{r'_{d,i} - x_{d,ij}(\beta_{d,j}^{(t)} - \beta_{d,j}^{(t+1)})\}_{i=1}^n\right\}_{d=1}^D$
    end
end
```

3.3 Tuning parameters

The implementation of SaENET(w) and GaLASSO in the *miselect* package sets tuning parameter values based on a 5-fold cross-validation routine combined with a “one-standard-error” rule (Hastie et al., 2009, 2015). More specifically, GaLASSO chooses the largest λ, such that the cross-validation error is within one standard error of the minimum cross-validation error. SaENET(w) selects an (α, λ) pair by first identifying all dyads whose cross-validation errors are within one standard error of the minimum cross-validation error, and then picking the (α, λ) pair with the largest L_1 penalty, i.e, $\lambda\alpha$. For both GaLASSO and SaENET(w), the default sequence for λ ranges from λ_{min} to λ_{max} on the log scale where λ_{max} is chosen to be the smallest value where all the coefficients are shrunk to zero, and $\lambda_{min} = 10^{-6}\lambda_{max}$. Since $\alpha \in [0, 1]$, then for SaENET(w) we can choose a sequence of values, say (0.2, 0.4, …, 0.8, 1), to fully explore the tradeoff between L_1 and L_2 shrinkage. When the non-adaptive methods are used, i.e. GLASSO, SENET(w), SLASSO(w), we set $\lambda_{min} = 10^{-3}\lambda_{max}$. The reason that λ_{min} is smaller for the adaptive methods is to prevent large adaptive weights from overwhelming the overall shrinkage.

One caveat with cross-validation to select tuning parameters for the stacked objective function approach is that, by stacking the imputed datasets on top of one another, there are now D rows corresponding to the same subject. Therefore, if the D rows corresponding to the same subject are distributed across the validation folds, then we are prone to overfitting. To prevent this issue, we restrict the fold assignment such that all of the rows corresponding to the same subject are assigned to a single validation fold.

4 Data example

Our motivating example is derived from data in the University of Michigan ALS Patient Biorepository which aims to identify environmental risk factors associated with ALS (Su et al., 2016; Goutman et al., 2019; Yu et al., 2014). ALS is progressive disease primarily involving motor neuron cells in the brain and spinal cord leading to weakness of voluntary muscles and death within 2-4 years due to respiratory failure (Goutman, 2017). ALS has a complex etiology driven by the combination of genetic susceptibility and environmental exposures (Paez-Colasante et al., 2015; Goutman et al., 2018; Al-Chalabi and Hardiman, 2013). In this particular study we are interested in characterizing the relationship between persistent organic pollutant (POP) exposure and ALS susceptibility. In total, 167 ALS cases and 99 healthy controls were recruited between 2011 and 2014. All participants provided written informed consent and the study was IRB approved (Su et al., 2016; Goutman et al., 2019). Participants provided their weight and height measurements at the time of study enrollment and 5 years prior to enrollment and their educational attainment. Plasma samples were collected from each study participant to measure 122 potentially neurotoxic POPs, which can be broadly partitioned into three chemical classes: organochlorine pesticides, polychlorinated biphenyls (PCBs), and polybrominated diphenyl ethers (PBDEs). Of the 122 POPs, a subset of 23 POPs with less than 60% non-detects were used for further analysis. More information regarding data collection and study protocols can be found in (Su et al., 2016; Goutman et al., 2019).

From a statistical perspective, the target model of interest is a penalized logistic regression model, where the outcome is ALS case/control status and the design matrix of predictors and covariates contains the 23 log-transformed POPs, age, sex, body mass index (BMI), rate of BMI change over the five years prior to survey consent, and education. The confounders are selected based on existing ALS literature (Chio et al., 2009). Elastic net regularization is of particular interest here, because many of the POPs have medium pairwise correlations with one another (Figure S1) (Goutman et al., 2019). We only penalize regression coefficients corresponding to POPs to ensure that adjustment covariates are retained in the final model. If we look at the percent missingness (Table S2), we observe that 9 of the 23 variables have more than 30% below the detection limit and 24.4% of subjects have incomplete BMI. Complete-case analysis in this context is infeasible, as every control is missing at least one covariate or has at least one measured POP below its respective detection limit. POP concentrations below their respective detection limits were imputed 10 or 50 times conditional on case/control status following the censored likelihood multiple imputation strategy outlined in Boss et al. (2019). After imputing the exposure non-detects, multiple imputation by chained equations (mice) was used to impute the missing adjustment covariates (Van Buuren and Groothuis-Oudshoorn, 2011).

To illustrate the problem with fitting separate penalized regression routines, we first apply LASSO, aLASSO, ENET, and aENET to each imputed dataset and calculate the proportion of imputed datasets in which each variable is selected. In Table 1, we summarize the results for scenarios with 10 and 50 imputed datasets. The final active set for each method is determined using the ad hoc rules outlined in Wood et al. (2008), namely (i) a variable is considered selected if it is selected in all imputed datasets, (ii) a variable is considered selected if it is selected in at least one imputed dataset, (iii) a variable is considered selected if it is selected in at least half of the imputed datasets. Note that depending on whether we use (i), (ii), or (iii), the number of selected POPs changes, especially when $D = 50$. For example, if ad hoc combining rule (i) is used, then ENET selects PBDE 153, pentachlorobenzene (PeCB), trans-chlordane, cis-nonachlor, and PCB 151. However, ad hoc combining rule (ii) additionally selects PBDE 28, PBDE 99, PCB 110, PCB 174, and PCB 180. As expected, the final active set determined by aLASSO and aENET is much more sparse than their non-adaptive counterparts; if ad hoc combining rule (iii) is used then aLASSO and aENET only select PeCB and cis-nonachlor.

A more subtle point that deserves further comment is that non-uniform POPs selection across imputed datasets makes it difficult to obtain final regression coefficient estimates. For example, consider aLASSO when $D = 50$, which selects PeCB in all 50 imputed datasets. Although PeCB is always selected, the interpretation of the regression coefficient for PeCB is conditional on the other selected exposures. That is, although PeCB is selected for all imputed datasets, the PeCB regression coefficient estimates are not necessarily comparable across imputed datasets because they condition on different sets of selected exposures.

The regression coefficient estimates obtained from the stacked and grouped objective function methods with 50 imputed datasets are in Table 2 and with 10 imputed datasets are in Table S3. Because the grouped methods produce different regression coefficient estimates across imputed datasets, the final estimates are the average of the regression coefficient estimates across imputed datasets. All coefficient estimates are positive, which is consistent with the hypothesis that higher POP exposure positively associates with a higher ALS risk. All of the non-adaptive methods select PBDE 153, PeCB, trans-chlordane, cis-nonachlor, and PCB 151. Similarly, the adaptive methods all select PeCB and cis-nonachlor, however the equally weighted stacked objective functions additionally select PCB 151 while the stacked objective functions with observational weights additionally select PBDE 153, trans-chlordane and PCB 151. Since all methods select PeCB and cis-nonachlor, we conclude that further studies should be conducted to assess the PeCB and cis-nonachlor neurotoxicity.

5 Simulation

We now evaluate the performance of the stacked methods and the grouped methods mentioned in Section 2, including SLASSO, SaLASSO, SENET, SaENET, SLASSO(w), SaLASSO(w), SENET(w), SaENET(w), GLASSO, and GaLASSO.

5.1 Simulation setting

We simulate 1000 datasets of size n containing outcome Y and p covariates $\boldsymbol{X}$ under 4 different cases. For Cases 1 and 2 we take $n = 500$ and $p = 20$, and for Cases 3 and 4 we take $n = 1000$ and $p = 100$. In all cases, covariates are generated from a multivariate normal distribution with zero mean and unit variance. The correlation structure of the covariates is block-diagonal, in order to mimic the correlation structure of the POPs. A more comprehensive breakdown of the correlation structure is detailed in Table 3.

Given $\boldsymbol{X}$, we generate a binary Y from

$$logit(P(Y = 1 \mid X)) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

The true value of $\boldsymbol{\beta}$ is specified according to different simulation cases, where Cases 1 and 3 correspond to concentrated signals and Cases 2 and 4 correspond to distributed signals. Here, signals are concentrated if there is only one non-null coefficient in a group, and signals are distributed if there is more than one non-null coefficient in a group. Regression coefficient magnitudes are set to fix the prevalence of $Y = 1$ at about 50% and maintain the Cox-Snell pseudo-R^2 at approximately 0.5. The four simulation settings are summarized in Table 3.

Missing values are generated under the Missing at Random (MAR) assumption (Little and Rubin, 2019). In all cases the outcome and the last covariate X_p are fully observed and the missingness indicator M_j for covariate X_j is generated from the logistic regression model

$$logit\left(Pr\left(M_j = 1\right)\right) = \alpha_{0j} + \alpha_{1j} X_p + \alpha_{2j} Y$$

where $R_j = 1$ indicates that covariate X_j is missing, and α_{0j}, α_{1j} and α_{2j} are chosen to control the percentage of missingness for X_j. For Case 1 and Case 2, about 25% of subjects are missing $\{X_1, \ldots, X_5\}$, 35% are missing $\{X_6, \ldots, X_{13}\}$, 45% are missing $\{X_{14}, \ldots, X_{17}\}$ and 55% are missing $\{X_{18}, X_{19}\}$. For Case 3 and Case 4, about 25% of subjects are missing $\{X_1, \ldots, X_{30}\}$, 35% are missing $\{X_{31}, \ldots, X_{60}\}$, 45% are missing $\{X_{61}, \ldots, X_{82}\}$, 55% are missing $\{X_{83}, \ldots, X_{95}\}$, and 60% are missing $\{X_{96}, \ldots, X_{99}\}$. In total, less than 5% of subjects have complete data, and about 13% of subjects have more than 90% data for all covariates. R package *mice* (Van Buuren and Groothuis-Oudshoorn, 2011) is used to multiply impute the missing data using predictive mean matching. Each simulated dataset is imputed 10 or 50 times. To obtain independent imputed values, we set the number of iterations to 30 in *mice*. The stacked and grouped methods are then applied to perform variable selection and parameter estimation.

5.2 Simulation results

We evaluate simulation results in terms of the following four metrics. In the following definitions, T and F are the number of non-null and null coefficients in the data generating model, respectively, R is the total number of simulation runs, $\hat{\beta}_j^r$ is the coefficient estimate for β_j in the r-th run. Since the estimates for the j-th coefficient by GLASSO and GaLASSO are different across imputed datasets, the mean $\frac{1}{D}\sum_{d=1}^{D}\hat{\beta}_{d,j}^{r}$ is used to approximate $\hat{\beta}_j^r$.

- Sensitivity (SENS) = $\frac{1}{RT}\sum_{r=1}^{R}(\#\, of\ selected\ non-null\ coefficients\ in\ the\ r_{th} run)$
- Specificity (SPEC) = $\frac{1}{RF}\sum_{r=1}^{R}(\#\, of\ selected\ null\ coefficients\ in\ the\ r_{th} run)$
- Mean squared error for non-null coefficients (MSE$_{\text{non-null}}$) = $\frac{1}{R}\sum_{r=1}^{R}\sum_{j=1}^{p}\left(\hat{\beta}_j^r-\beta_j\right)^2 I\left(\beta_j \neq 0\right)$
- Mean squared error for null coefficients (MSE$_{\text{null}}$) = $\frac{1}{R}\sum_{r=1}^{R}\sum_{j=1}^{p}\left(\hat{\beta}_j^r-\beta_j\right)^2 I\left(\beta_j = 0\right)$

Sensitivity and specificity capture the accuracy of variable selection and vary between 0 and 1, with larger values indicating better performance. The MSEs quantify the estimation accuracy with smaller values being preferred.

Figure 3–4 and Table S4 present sensitivity, specificity, MSE for non-null coefficients, and MSE for null coefficients for all four cases with 50 imputed datasets. Using SLASSO as the benchmark, the ratio to SLASSO for each measure is presented in Table S5. Overall, compared to the corresponding non-adaptive methods, the adaptive methods perform better with respect to estimation and selection. Specifically, the adaptive methods have similar sensitivity but considerably higher specificity, and considerably smaller MSE for non-null coefficients, except for GaLASSO under Case 4. For null coefficients, the adaptive methods have similar MSE to those of non-adaptive methods under Cases 1 and 2, and noticeably larger MSE under Cases 3 and 4, except for GaLASSO under Cases 3 and 4. Adaptive methods have high sensitivity across all cases and high specificity under Cases 1 and 2. Under Cases 3 and 4, GaLASSO has the highest specificity, followed by SaLASSO(w) and SaENET(w). The stacked methods with adaptive weights have smaller MSE for non-null coefficients than the grouped method with adaptive weights under all 4 cases. On the other hand, under Case 4, GaLASSO has the largest MSE for non-null coefficients and the smallest MSE for null coefficients, which is due to relatively low sensitivity and high specificity compared to other adaptive methods. In addition, the stacked approach performs better than the grouped approach in terms of estimation and selecting important variables. Compared to SLASSO, GLASSO has slightly lower sensitivity in all four cases, but has noticeably lower specificity in Cases 3 and 4. The MSE for non-null coefficients is larger for GLASSO than SLASSO in all four cases, in addition to the MSE for null coefficients in Cases 3 and 4, due to large variability in the coefficient estimates.

The average runtime for each method is presented in Table 4. Compared to the grouped methods, the stacked methods are faster in all cases. SLASSO is 5.7 times faster than GLASSO in Case 1 when the sample size is small. When the sample size is large, i.e. Case 3, SLASSO is 8.9 times faster than GLASSO. The adaptive methods have longer average runtimes than the non-adaptive methods because the adaptive methods require an elastic net initialization step to construct the adaptive weights.

To understand how the performance of these methods is affected by the number of imputed datasets, we provide simulation results for a smaller number of 10 imputed datasets in the supplementary materials and compare sensitivity, specificity, MSE for nonnull coefficients, and MSE for null coefficients (Figure S2–S3, Table S6–S7). Overall, increasing the number of imputed datasets improves the MSE for both non-null and null coefficients, but has little impact on sensitivity and specificity, especially when p is large. The improvement coming from an increased number of imputed datasets is more significant for the grouped methods than the stacked methods. For example, in Case 3, the MSE for non-null and null coefficients for SLASSO with 10 imputed datasets are 1.07 and 1.22, respectively, and with 50 imputed datasets are 1.04 and 1.08, respectively. While for GLASSO, the MSE for non-null and null coefficients with 10 imputed datasets are 4.09 and 42.27, respectively, and with 50 imputed datasets are 3.30 and 29.70, respectively.

6 Discussion

In this paper, we elucidated the difference between stacked and grouped pooled objective functions, which are both designed to achieve uniform variable selection across multiple imputed datasets. The stacked pooled objective function assumes that the underlying true signals are the same across imputed datasets, including the signal magnitude, while the grouped pooled objective function assumes uniform signal selection but allows for different active signal magnitudes across imputed datasets. We extended existing methods to handle binary outcomes, developed a MM algorithm combined with block coordinate descent updates to optimize grouped pooled objective functions for LASSO and aLASSO regularization, and derived cyclic coordinate descent algorithm for the stacked pooled objective functions with ENET and aENET regularization. Algorithms for implementing the stacked and grouped approaches outlined in Section 2, for both continuous and binary outcomes, are available in the *miselect* R package on CRAN.

From a practical perspective, there are several reasons that one might prefer stacked over grouped objective functions. Based on our simulations, the overall MSE for estimates generated by optimizing stacked objective functions was either smaller than or equal to the estimates generated by optimizing the grouped objective function, provided that adaptive weighting is used. We also observed that the total runtime for optimizing stacked pooled objective functions is noticeably lower compared to the grouped objective function optimization routine. An additional practical consideration is that the stacked objective functions are much easier to extend beyond ENET penalization. For example, if one wanted to use a hierarchical interaction detection penalty, such as hierNet (Bien et al., 2013), one would only need to download the *hierNet* package from CRAN, and use the existing hierNet implementation on the stacked imputed datasets. Conversely, grouped methods would necessitate the development of additional algorithms to optimize an objective function with both a group LASSO penalty and a hierarchical interaction detection penalty. Lastly, although we did not observe a substantial difference between equal weights and the observational weights proposed by Wan et al. (2015), there are still conceptual concerns with having the observational weights depend on the fraction of missingness. As we mentioned earlier, upweighting observations with more data artificially moves the analysis in the direction of complete-case analysis, which is known to be biased under MAR and MNAR missing data mechanisms.

The positive ALS-POPs associations identified in the data example add to a growing body of literature on environmental risk factors for ALS (Kamel et al., 2012; McGuire et al., 1997; Sutedja et al., 2008). A major advantage of the data collected in this study is that POP concentrations were measured in plasma samples, rather than through surveys. Since ALS results from the complex interplay of multiple risks combined with neurotoxic environmental exposures (the gene-time-environment hypothesis), we contend that additional work is needed to more fully understand gene and pesticide exposure interaction (Paez-Colasante et al., 2015; Goutman et al., 2018; Al-Chalabi and Hardiman, 2013).

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

This research was supported by the NSF DMS under Grant *1712933*; NIH under Grant *P30 CA 046591*; NIEHS under Grant *K23ES027221*; National ALS Registry/CDC/ATSDR CDCP-DHHS-US (CDC/ATSDR under Grant *200-2013-56856*); NIH/NINDS under Grant *R01NS127188*; NIH/NIEHS under Grant *R01ES030049*; NeuroNetwork for Emerging Therapies at Michigan Medicine, University of Michigan; Robert and Katherine Jacobs Environmental Health Initiative; Sinai Medical Staff Foundation and Scott L. Pranger. The authors would like to thank Alexander Rix, BS for software development support, Crystal Pacut, Jayna Duell, RN, Blake Swihart, and Daniel Burger for study support.

Declaration of interest statement

Dr. Stephen A. Goutman: scientific advisory to Biogen and ITF Pharma. DSMB service for Watermark Research Partners; additional research support from ALS Association.

References

Al-Chalabi A and Hardiman O (2013). The epidemiology of als: a conspiracy of genes, environment and time. Nature Reviews Neurology 9(11), 617. [PubMed: 24126629]

Bien J, Taylor J, and Tibshirani R (2013). A lasso for hierarchical interactions. The Annals of Statistics 41 (3), 1111–1141. [PubMed: 26257447]

Boss J, Mukherjee B, Ferguson KK, Aker A, Alshawabkeh AN, Cordero JF, Meeker JD, and Kim S (2019). Estimating outcome-exposure associations when exposure biomarker detection limits vary across batches. Epidemiology 30(5), 746–755. [PubMed: 31299670]

Breheny P and Huang J (2015). Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. Statistics and computing 25 (2), 173–187. [PubMed: 25750488]

Chen Q and Wang S (2013). Variable selection for multiply-imputed data with application to dioxin exposure study. Statistics in Medicine 32(21), 3646–3659. [PubMed: 23526243]

Chio A, Logroscino G, Hardiman O, Swingler R, Mitchell D, Beghi E, Traynor BG, Consortium E, et al. (2009). Prognostic factors in als: a critical review. Amyotrophic lateral sclerosis 10(5-6), 310–323. [PubMed: 19922118]

Friedman J, Hastie T, and Tibshirani R (2010). Regularization paths for generalized linear models via coordinate descent. Journal of statistical software 33(1), 1. [PubMed: 20808728]

Garcia RI, Ibrahim JG, and Zhu H (2010). Variable selection in the cox regression model with covariates missing at random. Biometrics 66(1), 97–104. [PubMed: 19459831]

Geronimi J and Saporta G (2017). Variable selection for multiply-imputed data with penalized generalized estimating equations. Computational statistics & data analysis 110, 103–114.

Ghosh D, Zhu Y, and Coffman DL (2015). Penalized regression procedures for variable selection in the potential outcomes framework. Statistics in Medicine 34 (10), 1645–1658. [PubMed: 25628185]

Goutman SA (2017). Diagnosis and clinical management of amyotrophic lateral sclerosis and other motor neuron disorders. CONTINUUM: Lifelong Learning in Neurology 23 (5), 1332–1359. [PubMed: 28968365]

Goutman SA, Boss J, Patterson A, Mukherjee B, Batterman S, and Feldman EL (2019). High plasma concentrations of organic pollutants negatively impact survival in amyotrophic lateral sclerosis. Journal of Neurology, Neurosurgery & Psychiatry 90(8), 907–912. [PubMed: 30760645]

Goutman SA, Chen KS, Paez-Colasante X, and Feldman EL (2018). Emerging understanding of the genotype–phenotype relationship in amyotrophic lateral sclerosis. In Handbook of clinical neurology, Volume 148, pp. 603–623. Elsevier. [PubMed: 29478603]

Hastie T and Qian J (2014). Glmnet vignette. Retrieve from http://www.web.Stanford.edu/~hastie/Papers/Glmnet_Vignette.pdf. Accessed September 20, 2016.

Hastie T, Tibshirani R, and Friedman J (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Hastie T, Tibshirani R, and Wainwright M (2015). Statistical learning with sparsity: the lasso and generalizations. Chapman and Hall/CRC.

Heymans MW, Van Buuren S, Knol DL, Van Mechelen W, and De Vet HC (2007). Variable selection under multiple imputation using the bootstrap in a prognostic study. BMC medical research methodology 7(1), 1–10. [PubMed: 17217545]

Johnson BA, Lin D, and Zeng D (2008). Penalized estimating functions and variable selection in semiparametric regression models. Journal of the American Statistical Association 103(482), 672–680. [PubMed: 20376193]

Kamel F, Umbach DM, Bedlack RS, Richards M, Watson M, Alavanja MCR, Blair A, Hoppin JA, Schmidt S, and Sandler DP (2012). Pesticide exposure and amyotrophic lateral sclerosis. NeuroToxicology 33(3), 457–462. [PubMed: 22521219]

Lachenbruch PA (2011). Variable selection when missing values are present: a case study. Statistical Methods in Medical Research 20 (4), 429–444. [PubMed: 20442196]

Little RJ and Rubin DB (2019). Statistical analysis with missing data, Volume 793. John Wiley & Sons.

Liu Y, Wang Y, Feng Y, and Wall MM (2016). Variable selection and prediction with incomplete high-dimensional data. Annals of Applied Statistics 10(1), 418–450. [PubMed: 27213023]

Long Q and Johnson BA (2015). Variable selection in the presence of missing data: resampling and imputation. Biostatistics 16(3), 596–610. [PubMed: 25694614]

McGuire V, Longstreth WT Jr., Nelson LM, Koepsell TD, Checkoway H, Morgan MS, and van Belle G (1997). Occupational exposures and amyotrophic lateral sclerosis. a population-based case-control study. American Journal of Epidemiology 145 (12), 1076–1088. [PubMed: 9199537]

Paez-Colasante X, Figueroa-Romero C, Sakowski SA, Goutman SA, and Feldman EL (2015). Amyotrophic lateral sclerosis: mechanisms and therapeutics in the epigenomic era. Nature Reviews Neurology 11 (5), 266. [PubMed: 25896087]

Robins JM and Wang N (2000). Inference for imputation estimators. Biometrika 87(1), 113–124.

Schafer JL (2001). Analyzing the nhanes iii multiply imputed data set: Methods and examples.

Su F-C, Goutman SA, Chernyak S, Mukherjee B, Callaghan BC, Batterman S, and Feldman EL (2016). Association of environmental toxins with amyotrophic lateral sclerosis. JAMA neurology 73 (7), 803–811. [PubMed: 27159543]

Sutedja NA, Veldink JH, Fischer K, Kromhout H, Heederik D, Huisman MH, Wokke JHJ, and van den Berg LH (2008). Exposure to chemicals and metals and risk of amyotrophic lateral sclerosis: A systematic review. Amyotrophic Lateral Sclerosis 10(5-6), 302–309.

Tibshirani R (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 58 (1), 267–288.

Van Buuren S and Groothuis-Oudshoorn K (2011). mice: Multivariate imputation by chained equations in r. Journal of Statistical Software 45(3), 1–67.

Wan Y, Datta S, Conklin DJ, and Kong M (2015). Variable selection models based on multiple imputation with an application for predicting median effective dose and maximum effect. Journal of Statistical Computation and Simulation 85(9), 1902–1916. [PubMed: 26412909]

Wang N and Robins JM (1998). Large-sample theory for parametric multiple imputation procedures. Biometrika 85 (4), 935–948.

Wood AM, White IR, and Royston P (2008). How should variable selection be performed with multiply imputed data? Statistics in Medicine 27(17), 3227–3246. [PubMed: 18203127]

Yang X, Belin TR, and Boscardin WJ (2005). Imputation and variable selection in linear regression models with missing covariates. Biometrics 61 (2), 498–506. [PubMed: 16011697]

Yang Y and Zou H (2015). A fast unified algorithm for solving group-lasso penalize learning problems. Statistics and Computing 25(6), 1129–1141.

Yang Y, Zou H, Yang MY, and Matrix D (2017). Package gcdnet.

Yu Y, Su F-C, Callaghan BC, Goutman SA, Batterman SA, and Feldman EL (2014). Environmental risk factors and amyotrophic lateral sclerosis (als): a case-control study of als in michigan. PloS one 9 (6).

Yuan M and Lin Y (2006). Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68(1), 49–67.

Zou H (2006). The adaptive lasso and its oracle properties. Journal of the American Statistical Association 101 (476), 1418–1429.

Zou H and Hastie T (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67(2), 301–320.

Zou H and Zhang HH (2009). On the adaptive elastic-net with a diverging number of parameters. The Annals of Statistics 37(4), 1733–1751. [PubMed: 20445770]

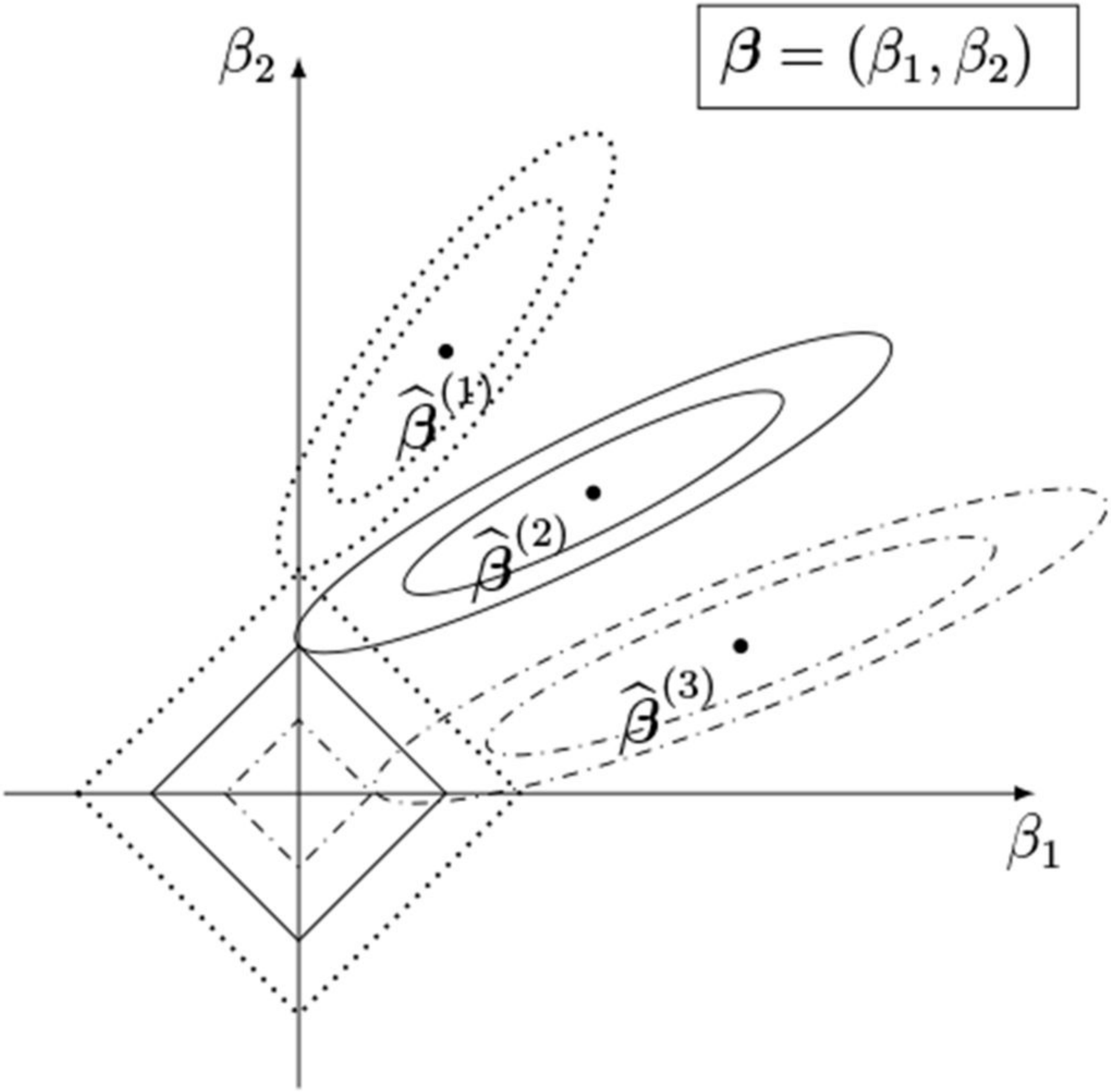


Figure 1:

Example of LASSO fit separately on each imputed dataset with two predictor variables. β_1 and β_2 represent the first and second coefficient, respectively. $\hat{\beta}^{(d)}$ denotes the maximum likelihood estimate in the d-th imputed dataset without the constraint ($d = 1, 2, 3$). The diamond-shaped areas around the origin represent the resulting constrained regions for the imputed datasets. The contours surrounding the maximum likelihood estimates represent the loss function.

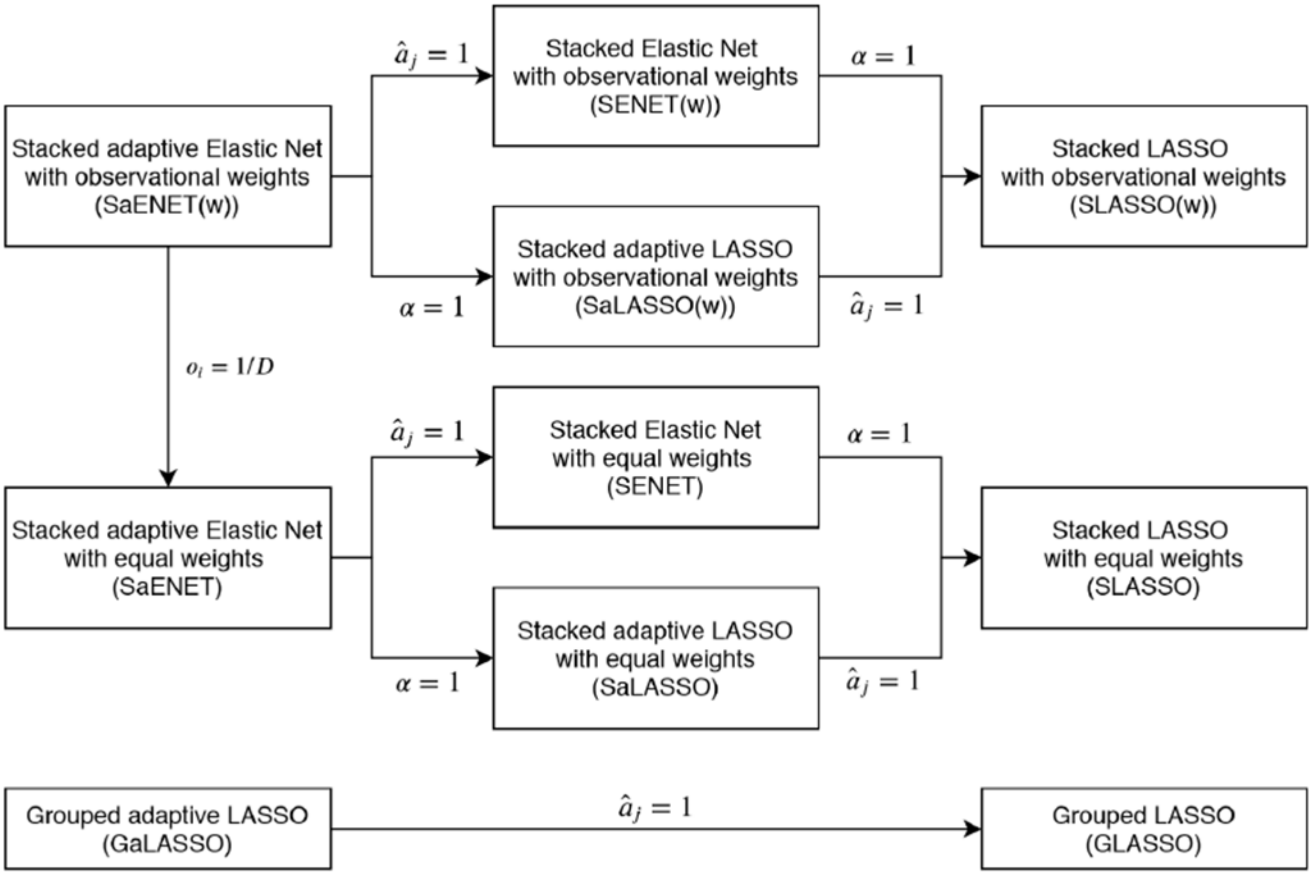

Stacked adaptive Elastic Net with observational weights (SaENET(w))
$\hat{a}_j = 1$
Stacked Elastic Net with observational weights (SENET(w))
$\alpha = 1$
Stacked adaptive LASSO with observational weights (SaLASSO(w))
$\alpha = 1$
$\hat{a}_j = 1$
Stacked LASSO with observational weights (SLASSO(w))
$o_i = 1/D$
Stacked adaptive Elastic Net with equal weights (SaENET)
$\hat{a}_j = 1$
Stacked Elastic Net with equal weights (SENET)
$\alpha = 1$
Stacked adaptive LASSO with equal weights (SaLASSO)
$\alpha = 1$
$\hat{a}_j = 1$
Stacked LASSO with equal weights (SLASSO)
Grouped adaptive LASSO (GaLASSO)
$\hat{a}_j = 1$
Grouped LASSO (GLASSO)

Figure 2:

Illustration of how to obtain penalized pooled objective function methods, which are special cases of SaENET(w) and GaLASSO.

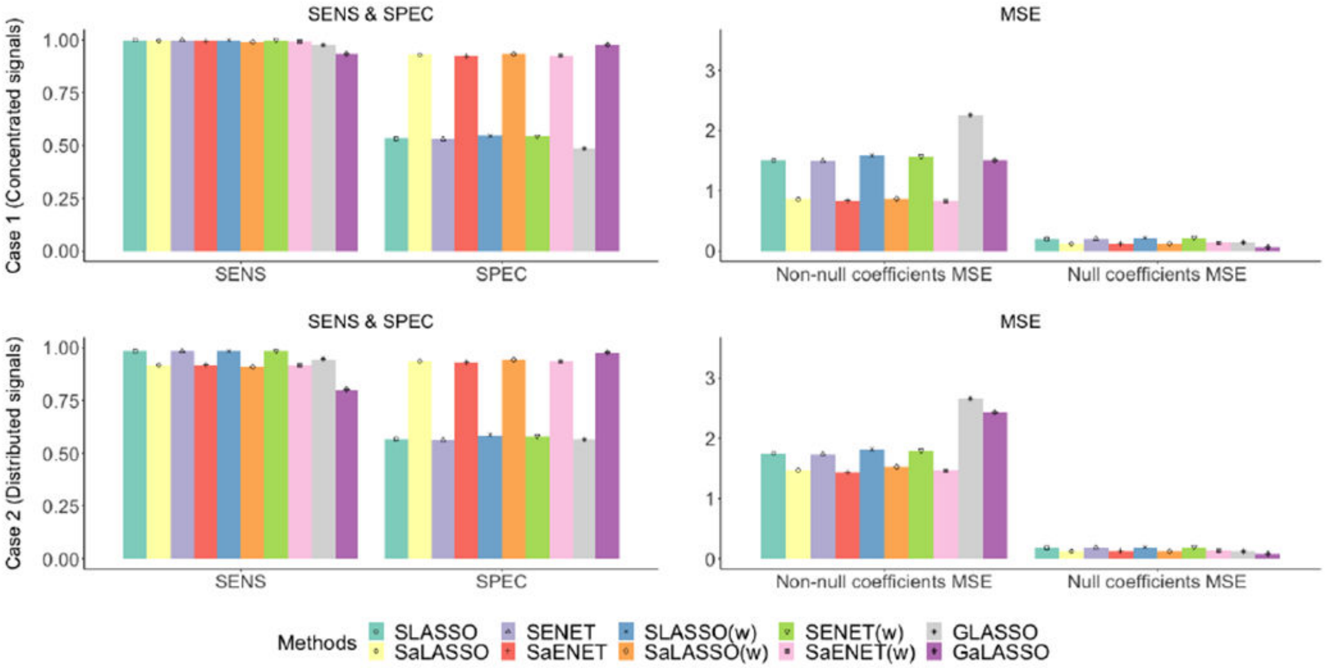


Figure 3:

Simulation results for Case 1 (top panel) and Case 2 (bottom panel) where n=500 and p=20 for 50 imputed datasets. Sensitivity (SENS) and specificity (SPEC) are on the left and MSE for non-null and null coefficients are on the right.

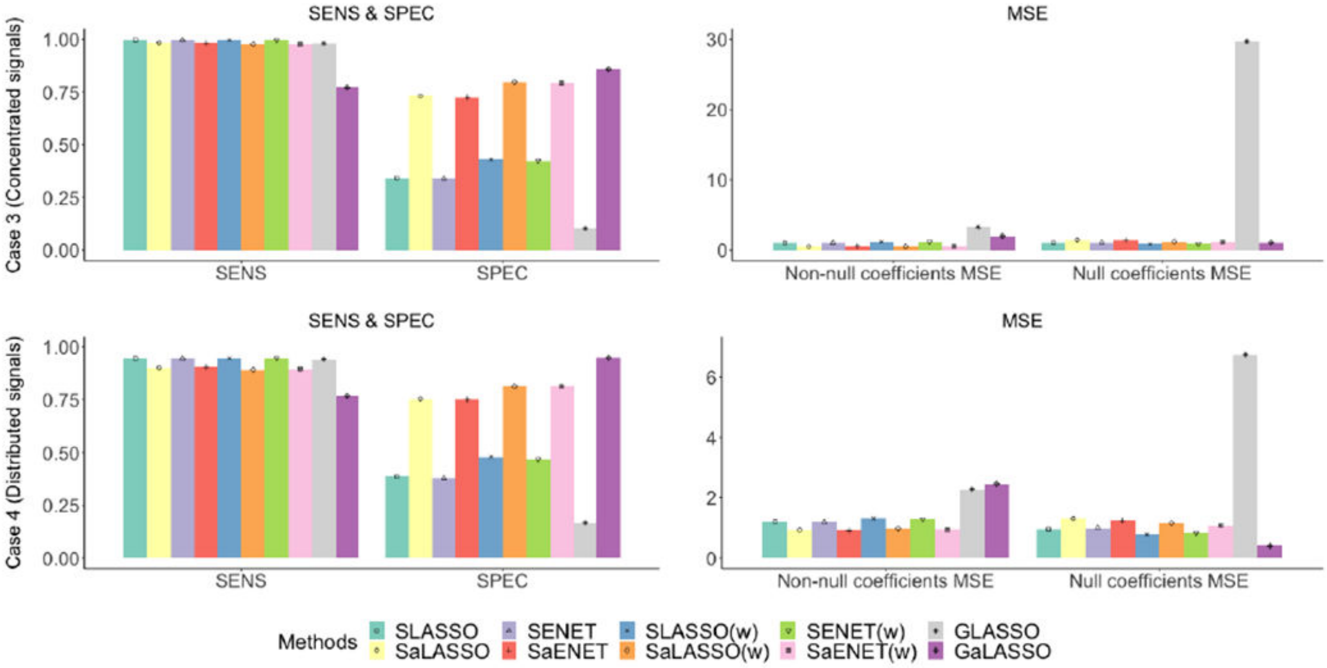


Figure 4:

Simulation results for Case 3 (top panel) and Case 4 (bottom panel) where n=1000 and p=100 for 50 imputed datasets. Sensitivity (SENS) and specificity (SPEC) are on the left and MSE for non-null and null coefficients are on the right.

Table 1:

The proportion of imputed datasets in which each POP is selected. Note that only 10 out of the 23 POPs are listed because the other 13 POPs were never selected by LASSO, aLASSO, ENET, or aENET. Bolded entries indicate a selection proportion over 0.50 and dashes denote a selection proportion of zero. D is the number of imputed datasets. The total number of POPs in the final active set, as determined by three ad hoc combining rules, are presented in the rows titled, Union (in at least one dataset), 50%-cutoff (in over 50% datasets), and Intersection (in all datasets).

POPs	**LASSO**		**aLASSO**		**ENET**		**aENET**	
	D=10	**D=50**	**D=10**	**D=50**	**D=10**	**D=50**	**D=10**	**D=50**
PBDE28	-	-	-	-	0.1	0.1	-	-
PBDE99	-	-	-	-	-	0.04	-	-
PBDE153	**1**	**1**	-	0.02	**1**	**1**	-	0.02
PeCB	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
trans-chlordane	**0.9**	**0.94**	0.1	0.08	**1**	**1**	0.1	0.08
cis-nonachlor	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
PCB 110	-	0.02	-	-	-	0.06	-	-
PCB 151	**0.9**	**0.96**	-	0.04	**1**	**1**	-	0.06
PCB 174	-	0.02	-	-	0.1	0.08	-	-
PCB 180	-	-	-	-	-	0.02	-	-
Union	5	7	3	5	7	10	3	5
50%-cutoff	5	5	2	2	5	5	2	2
Intersection	3	3	2	2	5	5	2	2

Table 2:

Regression coefficient estimates for five POPs collected as part of the University of Michigan ALS Patients Biorepository case-control study (167 ALS cases and 99 healthy controls). Results are based on 50 imputed datasets. Only five of the 23 POPs are displayed because the other 18 POPs were not selected by any method.

POPs	SLASSO	SaLASSO	SENET	SaENET	SLASSO (w)	SaLASSO (w)	SENET (w)	SaENET (w)	GLASSO	GaLASSO
PBDE 153	0.083	-	0.092	-	0.086	-	0.095	-	0.075	-
PeCB	0.331	0.744	0.367	0.715	0.284	0.670	0.322	0.657	0.299	0.968
trans-chlordane	0.074	-	0.109	-	0.073	0.034	0.109	0.036	0.044	-
cis-nonachlor	0.247	0.576	0.303	0.558	0.253	0.615	0.309	0.601	0.198	0.191
PCB151	0.080	0.156	0.128	0.151	0.057	0.189	0.111	0.184	0.038	-
# selected	**5**	**3**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**2**
# removed	**18**	**20**	**18**	**20**	**18**	**19**	**18**	**19**	**18**	**21**

Table 3:

Data generation details for all simulation settings. In the correlation structure column, covariates within the same parentheses have an exchangeable pairwise correlation structure with one another, but are independent from all other covariates. For Case 1 and Case 2, $n = 500$, $p = 20$, approximately 25% subjects are missing $\{X_1, \ldots, X_5\}$, 35% are missing $\{X_6, \ldots, X_{13}\}$, 45% are missing $\{X_{14}, \ldots, X_{17}\}$, and 55% are missing $\{X_{18}, X_{19}\}$. For Case 3 and Case 4, $n = 1000$, $p = 100$, about 25% subjects are missing $\{X_1, \ldots, X_{30}\}$, 35% are missing $\{X_{31}, \ldots, X_{60}\}$, 45% are missing $\{X_{61}, \ldots, X_{82}\}$, 55% are missing $\{X_{83}, \ldots, X_{95}\}$, and 60% are missing $\{X_{96}, \ldots, X_{99}\}$. The No. Signals (%) column refers to the number of covariates that have non-null coefficients and the corresponding relative percentage.

	Correlation structure	**Signal structure**	**No. Signals (%)**	**Signals**
Case 1	(X_1, X_2, X_3) as 0.9 (X_6, X_7, X_8) as 0.5 (X_{11}, X_{12}, X_{13}) as 0.3	Concentrated	5 (25%)	$\beta_1 = 2$, $\beta_4 = 1.5$, $\beta_7 = 1.5$, $\beta_{11} = 1$, $\beta_{14} = 1$;
Case 2	(X_1, X_2, X_3) as 0.9 (X_6, X_7, X_8) as 0.5 (X_{11}, X_{12}, X_{13}) as 0.3	Distributed	5 (25%)	$\beta_1 = 2$, $\beta_2 = 1$, $\beta_4 = 2$, $\beta_7 =1$, $\beta_{11} = 1$
Case 3	$(X_1, \ldots, X_6)$ as 0.9 $(X_{11}, \ldots, X_{16})$ as 0.5 $(X_{21}, \ldots, X_{26})$ as 0.3	Concentrated	10 (10%)	$\beta_2 = 2$, $\beta_7 = 0.8$, $\beta_9 = 0.8$, $\beta_{12} = 0.5$, $\beta_{17} = 1.5$, $\beta_{27} = 1$, $\beta_{37} = 0.8$, $\beta_{47} = 0.4$, $\beta_{48} = 1$, $\beta_{49} = 1$;
Case 4	$(X_1, \ldots, X_6)$ as 0.9 $(X_{11}, \ldots, X_{16})$ as 0.5 $(X_{21}, \ldots, X_{26})$ as 0.3	Distributed	10 (10%)	$\beta_1 = 1.2$, $\beta_2 = 0.8$, $\beta_3 = 0.4$, $\beta_4 = 0.4$, $\beta_{12} = 1.2$, $\beta_{13} = 1$ $\beta_{17} = 1.2$, $\beta_{27} = 1$, $\beta_{37}=1$, $\beta_{47} = 1$;

Table 4:

Average runtime (in minutes) of each method for 4 simulation cases with 10 imputed datasets (1000 replications). Case 1 and Case 2 have 500 observations and 20 covariates. Case 3 and Case 4 have 1000 observations and 100 covariates.

	SLASSO	SaLASSO	SENET	SaENET	SLASSO (w)	SaLASSO (w)	SENET (w)	SaENET (w)	GLASSO	GaLASSO
Case 1	12.8	56.7	51.3	77.1	12.7	52.4	47.7	69.1	73.1	86.1
Case 2	15.6	65.7	57.8	94.4	16.3	62.3	55.1	85.8	106.0	126.7
Case 3	127.0	585.0	557.8	704.5	129.6	549.2	527.9	638.2	1129.1	1182.8
Case 4	219.9	830.5	769.3	1086.5	194.7	736.4	686.3	931.4	1417.5	1467.0