Penalized Regression Methods for Interaction and Mixed-Effects Models with Applications to Genomic and Brain Imaging Data

Sahir Rai Bhatnagar

Doctor of Philosophy

Department of Epidemiology, Biostatistics and Occupational Health

McGill University
Montréal, Québec, Canada
July 2018

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

Chapter 1

Literature Review

The Literature Review is comprised of five sections. The first is a description of three general analytic strategies for high-dimensional data. The second and third sections describe two penalization methods that this thesis builds upon, namely the lasso and the group lasso. For each method we detail the algorithms used to fit these models and their convergence properties. In the fourth section we introduce penalized interaction models. This is followed by an introduction to penalized linear mixed models.

1.1 High-dimensional regression methods

In this thesis, we consider the prediction of an outcome variable y observed on n individuals from p variables, where p is much larger than n. Challenges in this high-dimensional context include not only building a good predictor which will perform well in an independent dataset, but also being able to interpret the factors that contribute to the predictions. This latter issue can be very challenging in ultra-high dimensional predictor sets. For example, multiple different sets of covariates may provide equivalent measures of goodness of fit (Fan et al., 2014), and therefore how does one decide which are important? With the advent of high-

throughput technologies in genomics and brain imaging studies, computational approaches to variable selection have become increasingly important. Broadly speaking, there are three main approaches to analyze high-dimensional data: 1) univariate regression followed by a multiple testing correction 2) multivariable penalized regression and 3) dimension reduction followed by a multivariable regression. We briefly introduce each of these analytic strategies below.

1.1.1 Univariate regression

Genome-wide association studies (GWAS) have become the standard method for analyzing genetic datasets. A GWAS consists of a series of univariate regressions followed by a multiple testing correction. This approach is simple and easy to implement, and has successfully identified thousands of genetic variants associated with complex diseases (https://www.genome.gov/gwastudies/). Despite these impressive findings, the discovered markers have only been able to explain a small proportion of the phenotypic variance known as the missing heritability problem (Manolio et al., 2009). One plausible explanation is that there are many causal variants that each explain a small amount of variation with small effect sizes (J. Yang et al., 2010). GWAS are likely to miss these true associations due to the stringent significance thresholds required to reduce the number of false positives (Manolio et al., 2009). Most statistical methods for performing multiple testing adjustments assume weak dependence among the variables being tested (Leek & Storey, 2008). Dependence among multiple tests can lead to incorrect Type 1 error rates (Lin et al., 2013) and highly variable significance measures (Leek & Storey, 2008). Even in the presence of weakly dependent variables, adjusting for multiple tests in whole genome studies can result in low power. Furthermore, the univariate regression approach does not allow for modeling the joint effect of many variants which may be biologically more plausible (Schadt, 2009). In the next section, we introduce multivariable penalized regression approaches which have been proposed to

address some of these limitations.

1.1.2 Multivariable penalized regression

For n observations and p covariates, consider the multiple linear regression model $\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{y}$ is a n-length response vector, $\mathbf{X}$ is an $n \times p$ design matrix, $\boldsymbol{\beta}$ is a p-length coefficient vector and $\boldsymbol{\varepsilon}$ is a n-length error vector. The least squares estimate is given by $\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\boldsymbol{y}$. In high-dimensional data, the problem is that $\mathbf{X}^T\mathbf{X}$ is singular because the number of covariates greatly exceeds the number of subjects. For example DNA microarrays measure the expression of approximately 20,000 genes. However, due to funding constraints, the sample size is often less than a few hundred. A common solution to this problem is through penalized regression, i.e., apply a constraint on the values of $\boldsymbol{\beta}$. The problem can be formulated as finding the vector $\boldsymbol{\beta}$ that minimizes the penalized sum of squares:

$$\underbrace{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j\right)^2}_{\text{goodness of fit}} + \underbrace{\sum_{j=1}^{p} p(\beta_j; \lambda, \gamma)}_{\text{penalty}} \tag{1.1}$$

The first term of (1.1) is the squared loss of the data and can be generalized to any loss function while the second term is a penalty which depends on non-negative tuning parameters λ and γ that control the amount of shrinkage to be applied to $\boldsymbol{\beta}$ and the degree of concavity of the penalty function, respectively. Several penalty terms have been developed in the literature. Ridge regression places a bound on the square of the coefficients (ℓ_2 penalty) (Hoerl & Kennard, 1970) which has the effect of shrinking the magnitude of the coefficients. This however does not produce parsimonious models as none of the coefficients can be shrunk to exactly 0. The Lasso (Tibshirani, 1996) overcomes this problem by placing a bound on the sum of the absolute values of the coefficients (ℓ_1 penalty) which sets some of them to 0, thereby simultaneously performing model selection. The Lasso, along with other forms of penalization (e.g. SCAD Fan & Li (2001), Fused Lasso (Tibshirani et al., 2005), Adaptive

Lasso (Zou, 2006), Relaxed Lasso (Meinshausen, 2007), MCP (Zhang, 2010)) have proven successful in many practical problems. Despite these encouraging results, such methods have low sensitivity in the presence of high empirical correlations between covariates because only one variable tends to be selected from the group of correlated or nearly linearly dependent variables (Bühlmann et al., 2013). As a consequence, there is rarely consistency on which variable is chosen from one dataset to another (e.g. in cross-validation folds). This behavior may not be well suited to certain genomic datasets in which large sets of predictors are highly correlated (e.g. a regulatory module) and are also associated with the response. The elastic net was proposed to benefit from the strengths of ridge regression's treatment of correlated variables and lasso's sparsity (Zou & Hastie, 2005). By placing both an ℓ_1 and ℓ_2 penalty on $\boldsymbol{\beta}$, the elastic net achieves model parsimony while yielding similar regression coefficients for correlated variables. These methods however do not take advantage of the grouping structure of the data. For example, cortical thickness measurements from magnetic resonance imaging (MRI) scans are often grouped into cortical regions of the Automated Anatomical Labelling (AAL) atlas (Tzourio-Mazoyer et al., 2002). Genes involved in the same cellular process (e.g. KEGG pathway (Kanehisa et al., 2008)) can also be placed into biologically meaningful groups. When regularizing with the ℓ_1 penalty, each variable is selected individually regardless of its position in the design matrix. Existing structures between the variables (e.g. spatial, networks, pathways) are ignored even though in many real-life applications the estimation can benefit from this prior knowledge in terms of both prediction accuracy and interpretability (Bach et al., 2012). The group lasso (Yuan & Lin, 2006) (and generalizations thereof) overcomes this problem by producing a structured sparsity (Bach et al., 2012), i.e., given a predetermined grouping of non-overlapping variables, all members of the group are either zero or non-zero. The main drawback when applying these methods to genomic data is that these groups may not be known *a priori*. Known pathways may not be relevant to the response of interest and the study of inferring gene networks is still in its infancy.

1.1.3 Dimension reduction together with regression

Due to the unknown grouping problem, several authors have suggested a two-step procedure where they first cluster or group variables in the design matrix and then subsequently proceed to model fitting where the feature space is some summary measure of each group. This idea dates back to 1957 when Kendall (Kendall, 1957) first proposed using principal components in regression. Hierarchical clustering based on the correlation of the design matrix has also been used to create groups of genes in microarray studies and for each level of hierarchy, the cluster average was used as the new set of potential predictors in forward-backward selection (Hastie et al., 2001) or the lasso (Park et al., 2007). Bühlmann et al. (2013) proposed a bottom-up agglomerative clustering algorithm based on canonical correlations and used the group lasso on the derived clusters. There are some advantages to these methods over the ones previously mentioned in Sections 1.1.1 and 1.1.2. First, the results may be more interpretable than the traditional lasso (and related methods) because the non-zero components of the prediction model represent sets of genes as opposed to individual ones. Second, by using genes which cluster well, we bias the inputs towards correlated sets of genes which are more likely to have similar function. Third, taking a summary measure of the resulting clusters can reduce the variance in prediction (overfitting) due to the compressed dimension of the feature space. Lastly, from a practical point of view this approach is flexible and easy to implement because efficient algorithms exist for both clustering (Müllner, 2013) and model fitting (Friedman et al., 2010; Y. Yang & Zou, 2014). A limitation of these approaches is that the clustering is done in an unsupervised manner, i.e., the clusters do not use the response information. This has the effect of assigning similar coefficient values to correlated features. Witten et al. (2014) proposed a method which encourages features that share an association with the response to take on similar coefficient values. This is useful in situations where only a fraction of the features in a cluster are associated with the response.

1.2 Lasso

Consider the multiple linear regression model $\boldsymbol{y} = \beta_0 + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{y} \in \mathbb{R}^n$ is the response, $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the design matrix, $\beta_0 \in \mathbb{R}$ is the intercept, $\boldsymbol{\beta} \in \mathbb{R}^p$ is the coefficient vector corresponding to $\mathbf{X}$ and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is a vector of iid random errors. For least-squares loss, the lasso estimator (Tibshirani, 1996; Zou, 2006) is defined as

$$\widehat{\boldsymbol{\beta}}(\lambda) = \underset{(\beta_0, \boldsymbol{\beta})}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} w_i (y_i - \beta_0 - (\mathbf{X}\boldsymbol{\beta})_i)^2 + \lambda \sum_{j=1}^{p} v_j |\beta_j| \tag{1.2}$$

where $(\mathbf{X}\boldsymbol{\beta})_i$ is the ith element of the n-length vector $\mathbf{X}\boldsymbol{\beta}$, $\lambda > 0$ is a tuning parameter which controls the amount of regularization, w_i is a known weight for the ith observation, and v_j is the penalty factor for the jth covariate. These penalty factors are known and allow parameters to be penalized differently. In particular, when $v_j = 1$ for $j = 1, \ldots, p$ then all parameters are regularized equally by λ, and when $v_j = 0$ the jth covariate is not penalized, i.e., it will always be included in the model. Note also that the intercept is not penalized. The estimator (1.2) simultaneously does variable selection and shrinks the regression coefficients towards 0. Depending on the choice of λ, $\widehat{\beta}_j(\lambda) = 0$ for some j's, and $\widehat{\beta}_j(\lambda)$ can be thought of as a shrunken least squares estimator (Bühlmann & Van De Geer, 2011). It is worth noting that (1.2) is a convex optimization problem and thus can be solved very efficiently using a block coordinate descent algorithm (Friedman et al., 2007; Tseng & Yun, 2009) for which we provide further details below. Other algorithms for solving this problem exist including LARS (Efron et al., 2004) and the homotopy algorithm (Osborne et al., 2000), but these have been largely preceded by the coordinate descent algorithm due to its speed and computational efficiency.

1.2.1 Block coordinate descent algorithm

In a series of seminal papers, Tseng lays the groundwork for a general purpose block coordinate descent algorithm (CGD) (Tseng, 2001; Tseng et al., 1988; Tseng & Yun, 2009) to minimize the sum of a smooth function f (i.e. continuously differentiable) and a separable convex function P of the form

$$Q_\lambda(\boldsymbol{\Theta}) = \arg\min_{\boldsymbol{\Theta}} \; f(\boldsymbol{\Theta}) + \lambda P(\boldsymbol{\Theta}) \tag{1.3}$$

At each iteration, the algorithm approximates $f(\boldsymbol{\Theta})$ in (1.3) by a strictly convex quadratic function and then applies block coordinate decent to generate a decent direction followed by an inexact line search along this direction (Tseng & Yun, 2009). For continuously differentiable $f(\cdot)$ and convex and block-separable $P(\cdot)$ (e.g. $P(\boldsymbol{\beta}) = \sum_i P_i(\beta_i)$), Tseng & Yun (2009) show that the solution generated by the CGD method is a stationary point of (1.3) if the coordinates are updated in a Gauss-Seidel manner, i.e., $Q_\lambda(\boldsymbol{\Theta})$ is minimized with respect to one parameter while holding all others fixed. The separability of the penalty function into a sum of functions of each individual parameter is the key to applying this algorithm to lasso type problems. Indeed, the CGD algorithm has been successfully applied in fixed effects models (Friedman et al., 2010; Meier et al., 2008) and linear mixed models (Schelldorfer et al., 2011). Following Tseng & Yun (2009), the general purpose CGD algorithm for solving (1.3) is given by Algorithm 1.

Algorithm 1: Coordinate Gradient Descent Algorithm to solve (1.3)

Set the iteration counter $k \leftarrow 0$ and choose initial values for the parameter vector $\boldsymbol{\Theta}^{(0)}$;

repeat

Approximate the Hessian $\nabla^2 f(\boldsymbol{\Theta}^{(k)})$ by a symmetric matix $H^{(k)}$:

$$H^{(k)} = \operatorname{diag}\left[\min\left\{\max\left\{\left[\nabla^2 f(\boldsymbol{\Theta}^{(k)})\right]_{jj}, 10^{-2}\right\} 10^{9}\right\}\right]_{j=1,\ldots,p} \tag{1.4}$$

for $j = 1, \ldots, p$ **do**

Solve the descent direction $d^{(k)} \coloneqq d_{H^{(k)}}(\Theta_j^{(k)})$;

$$d_{H^{(k)}}(\Theta_j^{(k)}) \leftarrow \arg\min_{d}\left\{\nabla f(\Theta_j^{(k)})d + \frac{1}{2}d^2 H_{jj}^{(k)} + \lambda P(\Theta_j^{(k)} + d)\right\} \tag{1.5}$$

Choose a stepsize;

$$\alpha_j^{(k)} \leftarrow \text{line search given by the Armijo rule}$$

Update;

$$\widehat{\Theta}_j^{(k+1)} \leftarrow \widehat{\Theta}_j^{(k)} + \alpha_j^{(k)} d^{(k)}$$

end

$k \leftarrow k + 1$

until *convergence criterion is satisfied*;

The Armijo rule is defined as follows (Tseng & Yun, 2009):

Choose $\alpha_{init}^{(k)} > 0$ and let $\alpha^{(k)}$ be the largest element of $\left\{\alpha_{init}^{k}\delta^{r}\right\}_{r=0,1,2,\ldots}$ satisfying

$$Q_\lambda(\Theta_j^{(k)} + \alpha^{(k)} d^{(k)}) \leq Q_\lambda(\Theta_j^{(k)}) + \alpha^{(k)} \varrho \Delta^{(k)} \tag{1.6}$$

where $0 < \delta < 1$, $0 < \varrho < 1$, $0 \leq \gamma < 1$ and

$$\Delta^{(k)} := \nabla f(\Theta_j^{(k)}) d^{(k)} + \gamma (d^{(k)})^2 H_{jj}^{(k)} + \lambda P(\Theta_j^{(k)} + d^{(k)}) - \lambda P(\Theta_j^{(k)}) \tag{1.7}$$

Common choices for the constants are $\delta = 0.1$, $\varrho = 0.001$, $\gamma = 0$, $\alpha_{init}^{(k)} = 1$ for all k (Bertsekas, 1999). In what follows, we use Algorithm 1 to solve the lasso estimator with least-squares loss given by (1.2). Without loss of generality, we assume the penalty factors (v_j) are all equal to 1.

Descent Direction

For simplicity, we remove the iteration counter (k) from the derivation below.

For $\Theta_j^{(k)} \in \{\beta_1, \ldots, \beta_p\}$, let

$$d_H(\Theta_j) = \arg\min_d G(d) \tag{1.8}$$

where

$$G(d) = \nabla f(\Theta_j) d + \frac{1}{2} d^2 H_{jj} + \lambda |\Theta_j + d|$$

Since $G(d)$ is not differentiable at $-\Theta_j$, we calculate the subdifferential $\partial G(d)$ and search for d with $0 \in \partial G(d)$:

$$\partial G(d) = \nabla f(\Theta_j) + d H_{jj} + \lambda u \tag{1.9}$$

where

$$u = \begin{cases} 1 & \text{if} \quad d > -\Theta_j \\ -1 & \text{if} \quad d < -\Theta_j \\ [-1, 1] & \text{if} \quad d = \Theta_j \end{cases} \tag{1.10}$$

We consider each of the three cases in (1.9) below

1. $d > -\Theta_j$

$$\partial G(d) = \nabla f(\Theta_j) + dH_{jj} + \lambda = 0$$
$$d = \frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}}$$

   Since $\lambda > 0$ and $H_{jj} > 0$, we have

$$\frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}} > \frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}} = d \overset{\text{def}}{>} -\Theta_j$$

   The solution can be written compactly as

$$d = \text{mid}\left\{\frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}}, -\Theta_j, \frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}}\right\}$$

   where $\text{mid}\{a, b, c\}$ denotes the median (mid-point) of a, b, c.

2. $d < -\Theta_j$

$$\partial G(d) = \nabla f(\Theta_j) + dH_{jj} - \lambda = 0$$
$$d = \frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}}$$

   Since $\lambda > 0$ and $H_{jj} > 0$, we have

$$\frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}} < \frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}} = d \overset{\text{def}}{<} -\Theta_j$$

Again, the solution can be written compactly as

$$d = \text{mid}\left\{\frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}}, -\Theta_j, \frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}}\right\}$$

3. $d_j = -\Theta_j$

   There exists $u \in [-1, 1]$ such that

$$\partial G(d) = \nabla f(\Theta_j) + dH_{jj} + \lambda u = 0$$
$$d = \frac{-(\nabla f(\Theta_j) + \lambda u)}{H_{jj}}$$

   For $-1 \leq u \leq 1$, $\lambda > 0$ and $H_{jj} > 0$ we have

$$\frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}} \leq d \stackrel{\text{def}}{=} -\Theta_j \leq \frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}}$$

   The solution can again be written compactly as

$$d = \text{mid}\left\{\frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}}, -\Theta_j, \frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}}\right\}$$

We see all three cases lead to the same solution for (1.8). Therefore the descent direction for $\Theta_j^{(k)} \in \{\beta_1, \ldots, \beta_p\}$ for the ℓ_1 penalty is given by

$$d = \text{mid}\left\{\frac{-(\nabla f(\beta_j) - \lambda)}{H_{jj}}, -\beta_j, \frac{-(\nabla f(\beta_j) + \lambda)}{H_{jj}}\right\} \tag{1.11}$$

Solution for the β parameter

If the Hessian $\nabla^2 f(\boldsymbol{\Theta}^{(k)}) > 0$ then $H^{(k)}$ defined in (1.4) is equal to $\nabla^2 f(\boldsymbol{\Theta}^{(k)})$. Using $\alpha_{init} = 1$, the largest element of $\left\{\alpha_{init}^{(k)}\delta^r\right\}_{r=0,1,2,\ldots}$ satisfying the Armijo Rule inequality is reached for $\alpha^{(k)} = \alpha_{init}^{(k)}\delta^0 = 1$. The Armijo rule update for the $\boldsymbol{\beta}$ parameter is then given

by

$$\beta_j^{(k+1)} \leftarrow \beta_j^{(k)} + d^{(k)}, \qquad j = 1, \ldots, p \tag{1.12}$$

Substituting the descent direction given by (1.11) into (1.12) we get

$$\beta_j^{(k+1)} = \text{mid}\left\{\beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)}) - \lambda)}{H_{jj}}, 0, \beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)}) + \lambda)}{H_{jj}}\right\} \tag{1.13}$$

We can further simplify this expression. First, we can re-write the loss function in (1.2) as

$$g(\boldsymbol{\beta}^{(k)}) = \frac{1}{2}\sum_{i=1}^{n} w_i \left(y_i - \sum_{\ell \neq j} X_{i\ell}\beta_\ell^{(k)} - X_{ij}\beta_j^{(k)}\right)^2 \tag{1.14}$$

The gradient and Hessian are given by

$$\nabla f(\beta_j^{(k)}) \coloneqq \frac{\partial}{\partial \beta_j^{(k)}} g(\boldsymbol{\beta}^{(k)}) = -\sum_{i=1}^{n} w_i X_{ij}\left(y_i - \sum_{\ell \neq j} X_{i\ell}\beta_\ell^{(k)} - X_{ij}\beta_j^{(k)}\right) \tag{1.15}$$

$$H_{jj} \coloneqq \frac{\partial^2}{\partial {\beta_j^{(k)}}^2} g(\boldsymbol{\beta}^{(k)}) = \sum_{i=1}^{n} w_i X_{ij}^2 \tag{1.16}$$

Substituting (1.15) and (1.16) into $\beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)}) - \lambda)}{H_{jj}}$

$$\begin{aligned}
&\beta_j^{(k)} + \frac{\sum_{i=1}^n w_i X_{ij}\left(y_i - \sum_{\ell \neq j} X_{i\ell}\beta_\ell^{(k)} - X_{ij}\beta_j^{(k)}\right) + \lambda}{\sum_{i=1}^n w_i X_{ij}^2} \\
&= \beta_j^{(k)} + \frac{\sum_{i=1}^n w_i X_{ij}\left(y_i - \sum_{\ell \neq j} X_{i\ell}\beta_\ell^{(k)}\right) + \lambda}{\sum_{i=1}^n w_i X_{ij}^2} - \frac{\sum_{i=1}^n w_i X_{ij}^2 \beta_j^{(k)}}{\sum_{i=1}^n w_i X_{ij}^2} \\
&= \frac{\sum_{i=1}^n w_i X_{ij}\left(y_i - \sum_{\ell \neq j} X_{i\ell}\beta_\ell^{(k)}\right) + \lambda}{\sum_{i=1}^n w_i X_{ij}^2}
\end{aligned} \tag{1.17}$$

Similarly, substituting (1.15) and (1.16) in $\beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)})+\lambda)}{H_{jj}}$ we get

$$\frac{\sum_{i=1}^n w_i X_{ij} \left(y_i - \sum_{\ell \neq j} X_{i\ell} \beta_\ell^{(k)} \right) - \lambda}{\sum_{i=1}^n w_i X_{ij}^2} \tag{1.18}$$

Finally, substituting (1.17) and (1.18) into (1.13) we get

$$\begin{aligned} \beta_j^{(k+1)} &= \text{mid} \left\{ \frac{\sum_{i=1}^n w_i X_{ij} \left(y_i - \sum_{\ell \neq j} X_{i\ell} \beta_\ell^{(k)} \right) - \lambda}{\sum_{i=1}^n w_i X_{ij}^2}, 0, \frac{\sum_{i=1}^n w_i X_{ij} \left(y_i - \sum_{\ell \neq j} X_{i\ell} \beta_\ell^{(k)} \right) + \lambda}{\sum_{i=1}^n w_i X_{ij}^2} \right\} \\ &= \frac{\mathcal{S}_\lambda \left(\sum_{i=1}^n w_i X_{ij} \left(y_i - \sum_{\ell \neq j} X_{i\ell} \beta_\ell^{(k)} \right) \right)}{\sum_{i=1}^n w_i X_{ij}^2} \end{aligned} \tag{1.19}$$

Where $\mathcal{S}_\lambda(x)$ is the soft-thresholding operator

$$\mathcal{S}_\lambda(x) = \text{sign}(x)(|x| - \lambda)_+$$

$\text{sign}(x)$ is the signum function

$$\text{sign}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$

and $(x)_+ = \max(x, 0)$. Since there is a closed form solution, which can be computed very quickly for each parameter update (given by (1.19)), the CGD algorithm is an attractive approach for solving the lasso estimator.

1.2.2 Lambda sequence

In general, the solution to (1.2) is computed over a decreasing sequence of values for the tuning parameter λ, beginning with the smallest value λ_{max} for which the entire coefficient vector $\widehat{\boldsymbol{\beta}} = \mathbf{0}_p$ (Friedman et al., 2010). To determine λ_{max}, we turn to the Karush-Kuhn-Tucker (KKT) optimality conditions for (1.2). These conditions can be written as

$$\begin{aligned} \frac{1}{v_j}\sum_{i=1}^{n} w_i X_{ij}\left(y_i - \sum_{j=1}^{p} X_{ij}\hat{\beta}_j\right) &= \lambda\gamma_j, \\ \gamma_j \in \begin{cases} \text{sign}(\hat{\beta}_j) & \text{if} \quad \hat{\beta}_j \neq 0 \\ [-1, 1] & \text{if} \quad \hat{\beta}_j = 0 \end{cases}, &\qquad \text{for } j = 1, \ldots, p \end{aligned} \tag{1.20}$$

where γ_j is the subgradient of the function $f(x) = |x|$ evaluated at $x = \hat{\beta}_j$. From (1.20), we can solve for the smallest value of λ such that the entire vector $(\hat{\beta}_1, \ldots, \hat{\beta}_p)$ is 0. This is given by

$$\lambda_{max} = \max_j \left\{ \left| \frac{1}{v_j}\sum_{i=1}^{n} w_i X_{ij} y_i \right| \right\}, \quad j = 1, \ldots, p \tag{1.21}$$

Following Friedman et al. (2010), we can choose $\tau\lambda_{max}$ to be the smallest value of tuning parameters λ_{min}, and construct a sequence of K values decreasing from λ_{max} to λ_{min} on the log scale. The defaults are set to $K = 100$, $\tau = 0.01$ if $n < p$ and $\tau = 0.001$ if $n \geq p$. The optimal value of λ can be chosen using 5-fold or 10-fold cross-validation. For least-squares loss, this corresponds to choosing the λ which minimizes the mean squared error.

1.2.3 Warm starts

The way in which we have derived the sequence of tuning parameters using the KKT conditions, allows us to exploit warm starts which has been shown to lead to computational speedups (Friedman et al., 2010). That is, the solution $\widehat{\boldsymbol{\Theta}}$ for λ_k is used as the initial value $\boldsymbol{\Theta}^{(0)}$ for λ_{k+1}.

1.2.4 Adaptive lasso

It has been shown that the lasso estimator can be produce biased estimates for large coefficients and give inconsistent variable selection results at the optimal λ for prediction, i.e., many noise features are included in the prediction model (Zou, 2006). To overcome the bias problems of the lasso, Zou (2006) proposed the adaptive lasso which allows a different amount of shrinkage for each regression coefficient using adaptive weights. Adaptive weighting has been shown to construct oracle procedures (Fan & Li, 2001), i.e., asymptotically, it performs as well as if the true model were given in advance. The adaptive lasso can be described as a two-stage procedure:

1. Calculate the initial regression estimates $\widehat{\boldsymbol{\beta}}_{init}$ from (1.2)
2. Refit (1.2) using penalty factors v_j equal to $1/\left|\hat{\beta}_{init,j}\right|$ for $j = 1, \ldots, p$.

As we can see from the weights, the adaptive lasso will shrink larger coefficients less which leads to consistent variable selection results under weaker conditions than the lasso (Bühlmann & Van De Geer, 2011). We detail the adaptive lasso procedure in Algorithm 2.

Algorithm 2: Adaptive lasso algorithm

1. For a decreasing sequence $\lambda = \lambda_{max}, \ldots, \lambda_{min}$, fit the lasso with $v_j = 1$ for $j = 1, \ldots, p$
2. Use cross-validation or a data splitting procedure to determine the optimal value for the tuning parameter: $\lambda^{[opt]} \in \{\lambda_{max}, \ldots, \lambda_{min}\}$
3. Let $\hat{\beta}^{[opt]}_{init,j}$ for $j = 1, \ldots, p$ be the coefficient estimates corresponding to the model at $\lambda^{[opt]}$
4. Set the weights to be $v_j = \left(\left|\hat{\beta}^{[opt]}_{init,j}\right|\right)^{-1}$ for $j = 1, \ldots, p$
5. Refit the lasso with the weights defined in step 4), and use cross-validation or a data splitting procedure to choose the optimal value of λ

1.3 Group Lasso

One main drawback of the lasso is that it ignores the grouping structure of the design matrix. When given a predetermined grouping of non-overlapping variables, we would want

all members of the group to be either zero or non-zero. For example, when dealing with categorical predictors where each factor is expressed through a set of indicator variables, removing an irrelevant factor is equivalent to setting the coefficients of the indicator variables to 0. In an additive model, where each variable is projected on to a set of basis function, e.g. $f_j(X_j) = \sum_{\ell=1}^{m_j} \psi_{j\ell}(X_j)\beta_{j\ell}$, we would want all $\{\beta_{j\ell}\}_{\ell=1}^{m_j}$ to be either zero or non-zero. This key difference between the lasso and group lasso penalty is shown in Figure **??**. Suppose we want to predict an individual's credit card balance from their age and height using the following additive model:

$$\text{credit card balance} = \beta_0 + \beta_{11}\text{age} + \beta_{12}\text{age}^2 + \beta_{21}\text{height} + \beta_{22}\text{height}^2 + \varepsilon \tag{1.22}$$

In Figures **??** and **??** we see that both the lasso and group lasso set the linear and quadratic terms for height $(\hat{\beta}_{21}, \hat{\beta}_{22})$ to 0. However, the lasso estimates only a nonzero quadratic term for age while the group lasso estimates both linear and quadratic terms to be nonzero.

We now provide details on the group lasso estimator. Assume that the predictors in the design matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$ belong to K groups and define the cardinality of index set I_k to be p_k. These groups are known *a priori* such that $(1, 2, \ldots, p) = \bigcup_{k=1}^{K} I_k$, and are also non-overlapping, i.e., $I_k \bigcap I_{k'} = \emptyset$ for $k \neq k'$. Therefore, group k contains p_k predictors corresponding to the columns of the design matrix X_j for $j \in I_k$, and $1 \leq k \leq K$. The intercept belongs to its own group, i.e., $I_1 = \{1\}$. The group lasso partitions the variable coefficients into K groups $\boldsymbol{\beta} = ([\boldsymbol{\beta}^{(1)}]^\intercal, [\boldsymbol{\beta}^{(2)}]^\intercal, \cdots, [\boldsymbol{\beta}^{(K)}]^\intercal)^\intercal$, where $\boldsymbol{\beta}^{(k)}$ denotes the segment of $\boldsymbol{\beta}$ corresponding to group k. For least-squares loss, the group lasso estimator (Yuan & Lin, 2006) is given by:

$$\widehat{\boldsymbol{\beta}}(\lambda) = \underset{(\beta_0,\boldsymbol{\beta})}{\arg\min} \frac{1}{2}\sum_{i=1}^{n} w_i(y_i - \beta_0 - (\mathbf{X}\boldsymbol{\beta})_i)^2 + \lambda \sum_{k=1}^{K} v_k \|\boldsymbol{\beta}^{(k)}\|_2 \tag{1.23}$$

where $\left\|\boldsymbol{\beta}^{(k)}\right\|_2 = \sqrt{\sum_{j\in I_k} \beta_j^2}$ and $\lambda > 0$ is the tuning parameter. As in the lasso estima-

tor (1.2), there are both observation weights w_i and penalty factors v_k where the latter is often set to $\sqrt{p_k}$ (Yuan & Lin, 2006). Solving the group lasso estimator is more challenging than the lasso since there is no closed form solution for (1.23). In the next section, we detail a majorization-minimization (MM) type algorithm (Lange et al., 2000; Y. Yang & Zou, 2015) used to solve (1.23).

1.3.1 Groupwise majorization descent algorithm

This description of the groupwise majorization descent (GMD) algorithm used to solve (1.23) follows mainly from Y. Yang & Zou (2015). The main difference here is that we consider a more general loss function of the form

$$L(\boldsymbol{\beta} \mid \mathbf{D}) = \frac{1}{2} [\boldsymbol{y} - \widehat{\boldsymbol{y}}]^\top \mathbf{W} [\boldsymbol{y} - \widehat{\boldsymbol{y}}] \tag{1.24}$$

where $\widehat{\boldsymbol{y}} = \hat{\beta}_0 + \mathbf{X}\widehat{\boldsymbol{\beta}}$, $\mathbf{D}$ is the working data $\{\boldsymbol{y}, \mathbf{X}\}$, and $\mathbf{W}$ is an $n \times n$ nonsingular and known weight matrix. This weight matrix can be used when the elements of $\boldsymbol{y}$ are correlated as is done in generalized least squares. The original proposal in Y. Yang & Zou (2015) is a special case of (1.24) where $\mathbf{W}$ is a diagonal matrix with entries equal to w_i. The loss function (1.24) satisfies the quadratic majorization (QM) condition, since $L(\boldsymbol{\beta} \mid \mathbf{D})$ is differentiable as a function of $\boldsymbol{\beta}$, i.e., $\nabla L(\boldsymbol{\beta}|\mathbf{D}) = -(\boldsymbol{y} - \hat{\boldsymbol{y}})^\top \mathbf{W}\mathbf{X}$, and there exists a $p \times p$ matrix $\mathbf{H} = \mathbf{X}^\top \mathbf{W}\mathbf{X}$ which only depends on the data $\mathbf{D}$, such that for all $\boldsymbol{\beta}, \boldsymbol{\beta}^*$,

$$L(\boldsymbol{\beta} \mid \mathbf{D}) \leq L(\boldsymbol{\beta}^* \mid \mathbf{D}) + (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \nabla L(\boldsymbol{\beta}^*|\mathbf{D}) + \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \mathbf{H}(\boldsymbol{\beta} - \boldsymbol{\beta}^*). \tag{1.25}$$

Noticing that the penalty term $\sum_{k=1}^K w_k ||\boldsymbol{\beta}_{(k)}||_2$ is separable with respect to the indices of the features $k = 1, \ldots, K$, we can derive the *groupwise-majorization-descent* (GMD) algorithm for computing the solution of (**??**) when the loss function satisfies the QM condition. Let $\widetilde{\boldsymbol{\beta}}$ denote the current solution of $\boldsymbol{\beta}$. Without loss of generality, let us derive the GMD update

of $\widetilde{\boldsymbol{\beta}}_{(k)}$, the coefficients of group k. Define $\mathbf{H}_k$ as the sub-matrix of $\mathbf{H}$ corresponding to group k. For example, if group 2 is $\{2, 4\}$ then $\mathbf{H}_{(2)}$ is a 2×2 matrix with

$$\mathbf{H}_{(2)} = \begin{bmatrix} h_{2,2} & h_{2,4} \\ h_{4,2} & h_{4,4} \end{bmatrix},$$

where $h_{i,j}$ is the i, jth entry of the $\mathbf{H}$ matrix. Write $\boldsymbol{\beta}$ such that $\boldsymbol{\beta}_{(k')} = \widetilde{\boldsymbol{\beta}}_{(k')}$ for $k' \neq k$. Given $\boldsymbol{\beta}_{(k')} = \widetilde{\boldsymbol{\beta}}_{(k')}$ for $k' \neq k$, the optimal $\boldsymbol{\beta}_{(k)}$ is defined as

$$\arg\min_{\boldsymbol{\beta}^{(k)}} L(\boldsymbol{\beta} \mid \mathbf{D}) + \lambda w_k \|\boldsymbol{\beta}_{(k)}\|_2. \tag{1.26}$$

Unfortunately, there is no closed form solution to (1.26) for a general loss function with general design matrix. We overcome the computational obstacle by taking advantage of the QM condition. From (1.25) we have

$$L(\boldsymbol{\beta} \mid \mathbf{D}) \leq L(\widetilde{\boldsymbol{\beta}} \mid \mathbf{D}) + (\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}})^\top \nabla L(\widetilde{\boldsymbol{\beta}}|\mathbf{D}) + \frac{1}{2}(\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}})^\top \mathbf{H}(\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}}).$$

Write $U(\widetilde{\boldsymbol{\beta}}) = -\nabla L(\widetilde{\boldsymbol{\beta}}|\mathbf{D})$. Using

$$\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}} = (\underbrace{0, \ldots, 0}_{k-1}, \boldsymbol{\beta}_{(k)} - \widetilde{\boldsymbol{\beta}}_{(k)}, \underbrace{0, \ldots, 0}_{K-k}),$$

we can write

$$L(\boldsymbol{\beta} \mid \mathbf{D}) \leq L(\widetilde{\boldsymbol{\beta}} \mid \mathbf{D}) - (\boldsymbol{\beta}_{(k)} - \widetilde{\boldsymbol{\beta}}_{(k)})^\top U_{(k)} + \frac{1}{2}(\boldsymbol{\beta}_{(k)} - \widetilde{\boldsymbol{\beta}}_{(k)})^\top \mathbf{H}_{(k)}(\boldsymbol{\beta}_{(k)} - \widetilde{\boldsymbol{\beta}}_{(k)}). \tag{1.27}$$

where

$$U_{(k)} = \frac{\partial}{\partial \boldsymbol{\beta}_{(k)}} L_Q(\boldsymbol{\beta} \mid \mathbf{D}) = -\left(Y - \hat{Y}\right)^\top \mathbf{W}\mathbf{X}_{(k)}, \tag{1.28}$$

$$\mathbf{H}_{(k)} = \frac{\partial^2}{\partial \boldsymbol{\beta}_{(k)} \partial \boldsymbol{\beta}_{(k)}^\top} L_Q(\boldsymbol{\beta} \mid \mathbf{D}) = \mathbf{X}_{(k)}^\top \mathbf{W}\mathbf{X}_{(k)}. \tag{1.29}$$

Let η_k be the largest eigenvalue of $\mathbf{H}_{(k)}$. We set $\gamma_k = (1 + \varepsilon^*)\eta_k$, where $\varepsilon^* = 10^{-6}$. Then we can further relax the upper bound in (1.27) as

$$L(\boldsymbol{\beta} \mid \mathbf{D}) \leq L(\widetilde{\boldsymbol{\beta}} \mid \mathbf{D}) - (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})^\top U_{(k)} + \frac{1}{2}\gamma_k (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})^\top (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)}). \tag{1.30}$$

It is important to note that the inequality strictly holds unless for $\boldsymbol{\beta}^{(k)} = \widetilde{\boldsymbol{\beta}}^{(k)}$. Instead of minimizing (1.26) we solve

$$\arg\min_{\boldsymbol{\beta}^{(k)}} L(\widetilde{\boldsymbol{\beta}} \mid \mathbf{D}) - (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})^\top U_{(k)} + \frac{1}{2}\gamma_k (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})^\top (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)}) + \lambda w_k \|\boldsymbol{\beta}^{(k)}\|_2. \tag{1.31}$$

Denote by $\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new})$ the solution to (1.31). It is straightforward to see that $\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new})$ has a simple closed-from expression

$$\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new}) = \frac{1}{\gamma_k}\left(U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\right)\left(1 - \frac{\lambda w_k}{\|U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\|_2}\right)_+. \tag{1.32}$$

Algorithm 3 summarizes the details of GMD.

Algorithm 3: The GMD algorithm for general group-lasso learning.

1. For $k = 1, \ldots, K$, compute γ_k, the largest eigenvalue of $\mathbf{H}^{(k)}$.
2. Initialize $\widetilde{\boldsymbol{\beta}}$.
3. Repeat the following cyclic groupwise updates until convergence:
   — for $k = 1, \ldots, K$, do step (3.1)–(3.3)
   3.1 Compute $U(\widetilde{\boldsymbol{\beta}}) = -\nabla L(\widetilde{\boldsymbol{\beta}}|\mathbf{D})$.
   3.2 Compute $\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new}) = \frac{1}{\gamma_k}\left(U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\right)\left(1 - \frac{\lambda w_k}{\|U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\|_2}\right)_+$.
   3.3 Set $\widetilde{\boldsymbol{\beta}}^{(k)} = \widetilde{\boldsymbol{\beta}}^{(k)}(\text{new})$.

1.3.2 Convergence

We can prove the strict descent property of GMD by using the MM principle (Hunter & Lange, 2004; Lange et al., 2000; Wu & Lange, 2010). Define

$$Q(\boldsymbol{\beta} \mid \mathbf{D}) = L(\widetilde{\boldsymbol{\beta}} \mid \mathbf{D}) - (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})^\top U^{(k)} + \frac{1}{2}\gamma_k (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)})^\top (\boldsymbol{\beta}^{(k)} - \widetilde{\boldsymbol{\beta}}^{(k)}) + \lambda w_k \|\boldsymbol{\beta}^{(k)}\|_2. \quad (1.33)$$

Obviously, $Q(\boldsymbol{\beta} \mid \mathbf{D}) = L(\boldsymbol{\beta} \mid \mathbf{D}) + \lambda w_k \|\boldsymbol{\beta}^{(k)}\|_2$ when $\boldsymbol{\beta}^{(k)} = \widetilde{\boldsymbol{\beta}}^{(k)}$ and (**??**) shows that $Q(\boldsymbol{\beta} \mid \mathbf{D}) > L(\boldsymbol{\beta} \mid \mathbf{D}) + \lambda w_k \|\boldsymbol{\beta}^{(k)}\|_2$ when $\boldsymbol{\beta}^{(k)} \neq \widetilde{\boldsymbol{\beta}}^{(k)}$. After updating $\widetilde{\boldsymbol{\beta}}^{(k)}$ using (**??**), we have

$$\begin{aligned} L(\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new}) \mid \mathbf{D}) + \lambda w_k \|\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new})\|_2 &\leq Q(\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new}) \mid \mathbf{D}) \\ &\leq Q(\widetilde{\boldsymbol{\beta}} \mid \mathbf{D}) \\ &= L(\widetilde{\boldsymbol{\beta}} \mid \mathbf{D}) + \lambda w_k \|\widetilde{\boldsymbol{\beta}}^{(k)}\|_2. \end{aligned}$$

Moreover, if $\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new}) \neq \widetilde{\boldsymbol{\beta}}^{(k)}$, then the first inequality becomes

$$L(\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new}) \mid \mathbf{D}) + \lambda w_k \|\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new})\|_2 < Q(\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new}) \mid \mathbf{D}).$$

Therefore, the objective function is strictly decreased after updating all groups in a cycle, unless the solution does not change after each groupwise update. If this is the case, we can

show that the solution must satisfy the KKT conditions, which means that the algorithm converges and finds the right answer. To see this, if $\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new}) = \widetilde{\boldsymbol{\beta}}^{(k)}$ for all k, then by the update formula (1.32) we have that for all k

$$\widetilde{\boldsymbol{\beta}}^{(k)} = \frac{1}{\gamma_k}\left(U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\right)\left(1 - \frac{\lambda w_k}{\|U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\|_2}\right) \qquad \text{if } \|U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\|_2 > \lambda w_k, \quad (1.34)$$

$$\widetilde{\boldsymbol{\beta}}^{(k)} = \mathbf{0} \qquad \text{if } \|U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\|_2 \leq \lambda w_k. \quad (1.35)$$

By straightforward algebra we obtain the KKT conditions:

$$-U^{(k)} + \lambda w_k \cdot \frac{\widetilde{\boldsymbol{\beta}}^{(k)}}{\|\widetilde{\boldsymbol{\beta}}^{(k)}\|_2} = \mathbf{0} \qquad \text{if } \widetilde{\boldsymbol{\beta}}^{(k)} \neq \mathbf{0},$$

$$\left\|U^{(k)}\right\|_2 \leq \lambda w_k \qquad \text{if } \widetilde{\boldsymbol{\beta}}^{(k)} = \mathbf{0},$$

where $k = 1, 2, \ldots, K$. Therefore, if the objective function stays unchanged after a cycle, the algorithm necessarily converges to the right answer.

1.4 Penalized interaction models

Then something about other structured L1 penalizations that have been proposed. There are quite a few examples of penalties built for specific applications, and maybe you could find a few such examples and cite them. Not comprehensively. Then you can point to the sail chapter as a new structured penalty.

Type	Method	Software
Linear	`CAP` (Zhao et al., 2009)	✗
	`SHIM` (Choi et al., 2010)	✗
	`hiernet` (Bien et al., 2013)	`hierNet(x, y)`
	`GRESH` (She & Jiang, 2014)	✗
	`FAMILY` (Haris et al., 2016)	`FAMILY(x, z, y)`
	`glinternet` (Lim & Hastie, 2015)	`glinternet(x, y)`
	`RAMP` (Hao et al., 2018)	`RAMP(x, y)`
	`LassoBacktracking` (Shah, 2016)	`LassoBT(x, y)`
Non-linear	`VANISH` (Radchenko & James, 2010)	✗
	`sail`	`sail(x, y, e)`

1.4.1 Current methods overview and their limitations

We consider a regression model for an outcome variable $Y = (y_1, \ldots, y_n)$ which follows an exponential family. Let $E = (e_1, \ldots, e_n)$ be the binary environment vector and $\boldsymbol{x} = (X_1, \ldots, X_p)$ be the matrix of high-dimensional data. Consider the regression model with main effects and their interactions with E:

$$g(\boldsymbol{\mu}) = \beta_0 + \underbrace{\beta_1 X_1 + \cdots + \beta_p X_p + \beta_E E}_{\text{main effects}} + \underbrace{\alpha_{1E}(X_1 E) + \cdots + \alpha_{pE}(X_p E)}_{\text{interactions}} \tag{1.36}$$

where $g(\cdot)$ is a known link function and $\boldsymbol{\mu} = \mathsf{E}[Y|\boldsymbol{x}, E, \boldsymbol{\beta}, \boldsymbol{\alpha}]$. Our goal is to estimate the parameters $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p, \beta_E) \in \mathbb{R}^{p+1}$ and $\boldsymbol{\alpha} = (\alpha_{1E}, \ldots, \alpha_{pE}) \in \mathbb{R}^p$ and to improve prediction of Y. In fact, in the light of our goals to improve prediction and interpretability,

we also consider the related model

$$g(\boldsymbol{\mu}) = \beta_0^* + \sum_{k=1}^{q} \beta_k^* \widetilde{X}_k + \beta_E^* E + \sum_{k=1}^{q} \alpha_k^* E \widetilde{X}_k \tag{1.37}$$

where $\widetilde{X}_k, k = 1, \ldots, q$ are linear combinations of X designed to reduce the dimension, such that $q << p$, and the superscript asterisk on the parameters is just to emphasize that these are different from those in (1.36). In what follows, we omit the asterisk on the parameters for clarity.

1.4.2 Phase 3: Variable Selection

We are interested in imposing the strong heredity principle (Chipman, 1996):

$$\hat{\alpha}_{jE} \neq 0 \qquad \Rightarrow \qquad \hat{\beta}_j \neq 0 \qquad \text{and} \qquad \hat{\beta}_E \neq 0 \tag{1.38}$$

In words, the interaction term will only have a non-zero estimate if its corresponding main effects are estimated to be non-zero. One benefit brought by hierarchy is that the number of measured variables can be reduced, referred to as practical sparsity Bien et al. (2013); She & Jiang (2014). For example, a model involving $X_1, E, X_1 \cdot E$ is more parsimonious than a model involving $X_1, E, X_2 \cdot E$, because in the first model a researcher would only have to measure two variables compared to three in the second model. In order to address these issues, we propose to extend the model of Choi *et al.* (Choi et al., 2010) to simultaneously perform variable selection, estimation and impose the strong heredity principle in the context of high dimensional interactions with the environment (HD$\times E$). To do so, we follow Choi and reparametrize the coefficients for the interaction terms as $\alpha_{jE} = \gamma_{jE}\beta_j\beta_E$. Plugging this into (1.36):

$$g(\boldsymbol{\mu}) = \beta_0 + \beta_1 \widetilde{X}_1 + \cdots + \beta_q \widetilde{X}_q + \beta_E E + \gamma_{1E}\beta_1\beta_E(\widetilde{X}_1 E) + \cdots + \gamma_{qE}\beta_q\beta_E(\widetilde{X}_q E) \tag{1.39}$$

where $\widetilde{\boldsymbol{x}} = (\widetilde{X}_1, \ldots, \widetilde{X}_q)$ are the cluster representatives derived in phase 2 and $q < p$. This reparametrization directly enforces the strong heredity principle (Eq. (1.38)), i.e., if either main effect estimates are 0, then $\hat{\alpha}_{jE}$ will be zero and a non-zero interaction coefficient implies non-zero $\hat{\beta}_j$ and $\hat{\beta}_E$. To perform variable selection in this new parametrization, we follow Choi *et al.* Choi et al. (2010) and penalize $\boldsymbol{\gamma} = (\gamma_{1E}, \ldots, \gamma_{pE})$ instead of penalizing $\boldsymbol{\alpha}$ as in (**??**), leading to the following penalized least squares criterion:

$$\underset{\beta_0, \boldsymbol{\beta}, \boldsymbol{\gamma}}{\arg\min} \frac{1}{2}\|Y - g(\boldsymbol{\mu})\|^2 + \lambda_\beta \left(w_1\beta_1 + \cdots + w_q\beta_q + w_E\beta_E\right) + \lambda_\gamma \left(w_{1E}\gamma_{1E} + \cdots + w_{qE}\gamma_{qE}\right) \tag{1.40}$$

where $g(\boldsymbol{\mu})$ is from (1.39), λ_β and λ_γ are tuning parameters and $\mathbf{w} = (w_1, \ldots, w_q, w_{1E}, \ldots, w_{qE})$ are prespecified adaptive weights. The λ_β tuning parameter controls the amount of shrinkage applied to the main effects, while λ_γ controls the interaction estimates and allows for the possibility of excluding the interaction term from the model even if the corresponding main effects are non-zero. It can be shown that the procedure in (1.40) asymptotically possesses the oracle property (Choi et al., 2010), even when the number of parameters tends to ∞ as the sample size increases, if the weights are chosen such that

$$w_j = \left|\frac{1}{\hat{\beta}_j}\right|, \quad w_{jE} = \left|\frac{\hat{\beta}_j\hat{\beta}_E}{\hat{\alpha}_{jE}}\right| \quad \text{for } j = 1, \ldots, q \tag{1.41}$$

where $\hat{\beta}_j$ and $\hat{\alpha}_j$ are the MLEs, *using the transformed variables*, from (1.36) or the ridge regression estimates when $q > n$. The rationale behind the data-dependent $\hat{\boldsymbol{w}}$ is that as the sample size grows, the weights for the truly zero predictors go to ∞ (which translates to a large penalty), whereas the weights for the truly non-zero predictors converge to a finite constant (Zou, 2006).

There have been several more recent proposals for modeling interactions with the strong heredity constraint in the variable selection via penalization literature including Composite Absolute Penalties (CAP) (Zhao et al., 2009), Variable selection using Adaptive Nonlinear

Interaction Structures in High dimensions (VANISH) (Radchenko & James, 2010), Strong Hierarchical Lasso (hierNet) (Bien et al., 2013), Group-Lasso Interaction Network (glinternet) (Lim & Hastie, 2015), Group Regularized Estimation under Structural Hierarchy (GRESH) (She & Jiang, 2014) and a Framework for Modeling Interactions with a Convex Penalty (FAMILY) (Haris et al., 2014). While each method has their own merit, including that they are all convex optimization problems, they all contain complex penalty functions which are hard to interpret and lead to computationally expensive fitting algorithms. On the other hand, the objective function in (1.40) can be solved using an iterative approach (by first fixing $\boldsymbol{\beta}$ and then $\boldsymbol{\alpha}$) which simplifies to a LASSO type problem; one that has been extensively studied, is well understood and can be solved efficiently using existing software (e.g. `glmnet` (Friedman et al., 2010)). A limitation of this approach is that the optimization problem is non-convex, arising from the reparametrization of $\boldsymbol{\alpha}$ as a product of optimization variables $(\boldsymbol{\beta}, \boldsymbol{\gamma})$, and hence convergence to the global minimum is not guaranteed (Choi et al., 2010). We argue that since there is only one E, and that $\widetilde{X}$ is much smaller in dimension, finding a solution is much more likely.

To our knowledge, strong hierarchies have never previously been used in HD interaction analysis in genomics or brain imaging studies. Furthermore, the specific choices of weights proposed here, i.e., based on the transformed variables from phase 2, have not been previously used. Choi *et al.* (Choi et al., 2010) estimated their weights simultaneously, but this would not be feasible in HD data. Finally, the adaptation to interactions with one key E variable is specific to our situation and this leads to computational efficiencies. These three points constitute novel aspects of this thesis. I have a working implementation of this, and am in the process of conducting simulation studies.

1.5 Penalized linear mixed models

(5) A brief intro to linear mixed models followed by why naive penalization violates the normality of residuals to motivate your ggmix chapter.

References

Bach, F., Jenatton, R., Mairal, J., Obozinski, G., et al. (2012). Structured sparsity through convex optimization. *Statistical Science*, *27*(4), 450–468.

5

Bertsekas, D. P. (1999). *Nonlinear programming*. Athena scientific Belmont.

10

Bien, J., Taylor, J., Tibshirani, R., et al. (2013). A lasso for hierarchical interactions. *The Annals of Statistics*, *41*(3), 1111–1141.

23, 24, 26

Bühlmann, P., Rütimann, P., van de Geer, S., & Zhang, C.-H. (2013). Correlated variables in regression: clustering and sparse estimation. *Journal of Statistical Planning and Inference*, *143*(11), 1835–1858.

5, 6

Bühlmann, P., & Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

7, 16

Chipman, H. (1996). Bayesian variable selection with related predictors. *Canadian Journal of Statistics*, *24*(1), 17–36.

24

Choi, N. H., Li, W., & Zhu, J. (2010). Variable selection with the strong heredity constraint and its oracle property. *Journal of the American Statistical Association*, *105*(489), 354–364.

23, 24, 25, 26

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of statistics*, *32*(2), 407–499.

7

Fan, J., Han, F., & Liu, H. (2014). Challenges of big data analysis. *National science review*, *1*(2), 293–314.

2

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, *96*(456), 1348–1360.

4, 16

Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., et al. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, *1*(2), 302–332.

7

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, *33*(1), 1.

6, 8, 15, 26

Hao, N., Feng, Y., & Zhang, H. H. (2018). Model selection for high-dimensional quadratic regression via regularization. *Journal of the American Statistical Association*, 1–11.

23

Haris, A., Witten, D., & Simon, N. (2014). Convex modeling of interactions with strong heredity. *arXiv preprint arXiv:1410.3517*.

26

Haris, A., Witten, D., & Simon, N. (2016). Convex modeling of interactions with strong heredity. *Journal of Computational and Graphical Statistics*, *25*(4), 981–1004.

23

Hastie, T., Tibshirani, R., Botstein, D., & Brown, P. (2001). Supervised harvesting of expression trees. *Genome Biology*, *2*(1), 1–0003.

6

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

4

Hunter, D., & Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician*, *58*(1), 30–37.

21

Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., ... others (2008). Kegg for linking genomes to life and the environment. *Nucleic acids research*, *36*(suppl 1), D480–D484.

5

Kendall, M. (1957). *A course in multivariate analysis*. London: Griffin.

6

Lange, K., Hunter, D., & Yang, I. (2000). Optimization transfer using sur- rogate objective functions (with discussion). *Journal of Computational and Graphical Statistics*, *9*, 1-20.

18, 21

Leek, J. T., & Storey, J. D. (2008). A general framework for multiple testing dependence. *Proceedings of the National Academy of Sciences*, *105*(48), 18718–18723.

3

Lim, M., & Hastie, T. (2015). Learning interactions via hierarchical group-lasso regularization. *Journal of Computational and Graphical Statistics*, *24*(3), 627–654.

23, 26

Lin, X., Lee, S., Christiani, D. C., & Lin, X. (2013). Test for interactions between a genetic marker set and environment in generalized linear models. *Biostatistics*, kxt006.

3

Manolio, T. A., Collins, F. S., Cox, N. J., Goldstein, D. B., Hindorff, L. A., Hunter, D. J., ... others (2009). Finding the missing heritability of complex diseases. *Nature*, *461*(7265), 747–753.

3

Meier, L., Van De Geer, S., & Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *70*(1), 53–71.

8

Meinshausen, N. (2007). Relaxed lasso. *Computational Statistics & Data Analysis*, *52*(1), 374–393.

5

Müllner, D. (2013). fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python. *Journal of Statistical Software*, *53*(9), 1–18. Retrieved from http://www.jstatsoft.org/v53/i09/

6

Osborne, M. R., Presnell, B., & Turlach, B. A. (2000). A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, *20*(3), 389–403.

7

Park, M. Y., Hastie, T., & Tibshirani, R. (2007). Averaged gene expressions for regression. *Biostatistics*, *8*(2), 212–227.

6

Radchenko, P., & James, G. M. (2010). Variable selection using adaptive nonlinear interaction structures in high dimensions. *Journal of the American Statistical Association*, *105*(492), 1541–1553.

23, 26

Schadt, E. E. (2009). Molecular networks as sensors and drivers of common human diseases. *Nature*, *461*(7261), 218–223.

3

Schelldorfer, J., Bühlmann, P., DE, G., & VAN, S. (2011). Estimation for high-dimensional linear mixed-effects models using l1-penalization. *Scandinavian Journal of Statistics*, *38*(2), 197–214.

8

Shah, R. D. (2016). Modelling interactions in high-dimensional data with backtracking. *Journal of Machine Learning Research*, *17*(207), 1–31.

23

She, Y., & Jiang, H. (2014). Group regularized estimation under structural hierarchy. *arXiv preprint arXiv:1411.4691*.

23, 24, 26

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

4, 7

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(1), 91–108.

4

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, *109*(3), 475–494.
8

Tseng, P., et al. (1988). Coordinate ascent for maximizing nondifferentiable concave functions. 8

Tseng, P., & Yun, S. (2009). A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, *117*(1), 387–423.
7, 8, 10

Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., ... Joliot, M. (2002). Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *Neuroimage*, *15*(1), 273–289.
5

Witten, D. M., Shojaie, A., & Zhang, F. (2014). The cluster elastic net for high-dimensional regression with unknown variable grouping. *Technometrics*, *56*(1), 112–122.
6

Wu, T., & Lange, K. (2010). The MM alternative to EM. *Statistical Science*, *4*, 492–505.
21

Yang, J., Benyamin, B., McEvoy, B. P., Gordon, S., Henders, A. K., Nyholt, D. R., ... others (2010). Common snps explain a large proportion of the heritability for human height. *Nature genetics*, *42*(7), 565.
3

Yang, Y., & Zou, H. (2014). gglasso: Group lasso penalized learning using a unified bmd

algorithm. Retrieved from http://CRAN.R-project.org/package=gglasso (R package version 1.3)

6

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

18

Yuan, M., & 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.

5, 17, 18

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 894–942.

5

Zhao, P., Rocha, G., & Yu, B. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, 3468–3497.

23, 25

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

5, 7, 16, 25

Zou, H., & 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.

5