Chapter 1

Introduction

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 (HD) 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?

When $p >> n$, standard generalized linear models (GLMs) methodology cannot uniquely estimate the unknown coefficient vector β. Moreover, even when $p \leq n$ with p close to n, standard errors of GLMs are likely to be inflated and parameter estimates unstable (Reid et al., 2016). In these instances it may be useful to assume the *Bet on Sparsity Principle* which says:

> Use a procedure that does well in sparse problems, since no procedure does well in dense problems (Friedman et al., 2001).

In fact, sometimes this assumption is necessary since we often don't have a large enough sample size to estimate so many parameters. Even when we do, we might want to identify a

remove any contractions "don't" "do not"

relatively small number of predictors that play an important role on the response. Applied researchers in the health sciences might prefer a simpler model because it can shed light on the etiology of disease. The sparsity assumption may also result in faster computations and more stable predictions on new datasets.

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 such HD data: 1) univariate regression followed by a multiple testing correction 2) multivariable penalized regression and 3) dimension reduction followed by a multivariable regression.

In this thesis, we focus on the use of penalized regression methods for variable selection and prediction in HD settings. In Chapter 2, we provide a critical review of the literature on penalized regression methods and the computational algorithms used to fit these models. In Chapter 3, we develop the sparse additive interaction learning model `sail` for detecting non-linear interactions with a key environmental or exposure variable in HD data. In Chapter 4, we develop a general penalized linear mixed model framework called `ggmix` that simultaneously selects and estimates variables while accounting for between individual correlations in one step. Chapter 5 explores whether use of exposure-dependent clustering relationships in dimension reduction can improve predictive modelling in a two-step framework that we develop called `eclust`. Chapters 3, 4 and 5 were originally written as stand-alone papers and as a result, there is some inconsistency in notation. Chapter 5 has been published in *Genetic Epidemiology*. Chapters 3 and 4 will be submitted to a statistical journal shortly after the submission of the thesis. We have published open source and freely available `R` packages for each of the methods developed Chapters 3, 4 and 5 (available at `https://github.com/sahirbhatnagar/`). Table 1.1 provides an overview of our software packages including some of their key characteristics. In Chapter 6, we conclude with an overview of the three manuscripts.

Table 1.1: Overview of Our Software Packages

	`eclust`	`sail`	`ggmix`
Model			
Least-Squares	✓	✓	✓
Binary Classification	✓		
Survival Analysis			
Penalty			
Ridge	✓		✓
Lasso	✓	✓	✓
Elastic Net	✓		✓
Group Lasso		✓	✓
Feature			
Interactions	✓	✓	
Flexible Modeling	✓	✓	
Random Effects			✓
Data	(x, y, e)	(x, y, e)	(x, y, Ψ)

Add a more detailed legend describing the headers "Model" "Penalty" and "Feature". Define (x, y, e, Ψ). Refer reader to chapter 2 in this legend.

Chapter 2

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 a brief introduction to linear mixed-effects models.

2.1 High-dimensional regression methods

We briefly introduce three of the main analytic strategies used to analyze HD data below.

2.1.1 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.

2.1.2 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:

$$\underbrace{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j\right)^2}_{\text{goodness of fit}} + \underbrace{\sum_{j=1}^{p} p(\beta_j; \lambda, \gamma)}_{\text{penalty}} \tag{2.1}$$

The first term of (2.1) 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.

2.1.3 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 2.1.1 and 2.1.2. 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).

In this context, we introduce a new two-step procedure called `eclust` (Bhatnagar et al., 2018) in Chapter 5 of the thesis. Our method is motivated by the fact that exposure variables (e.g. smoking) can alter correlation patterns between clusters of high-dimensional variables, i.e., alter network properties of the variables. However, it is not well understood whether such altered clustering is informative in prediction. In this paper, we explore whether use of exposure-dependent clustering relationships in dimension reduction can improve predictive modelling in a two-step framework.

A limitation of two-step methods 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.

Something is missing

★ Discuss sensitivity in 2.1.3

Add a connector sentence:

eg "Now that these three general strategies for predictive modelling with $p >> n$ have been described, the next section describes several specialized methods in detail."

2.2 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

$$\widehat{\boldsymbol{\beta}}(\lambda) = \underset{(\beta_0, \boldsymbol{\beta})}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} w_i (y_i - \beta_0 - (\mathbf{X}\boldsymbol{\beta})_i)^2 + \lambda \sum_{j=1}^{p} v_j |\beta_j| \tag{2.2}$$

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 (2.2) 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 (2.2) 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.

2.2.1 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

$$Q_\lambda(\Theta) = \arg\min_{\Theta} \; f(\Theta) + \lambda P(\Theta) \tag{2.3}$$

At each iteration, the algorithm approximates $f(\Theta)$ in (2.3) 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 (2.3) if the coordinates are updated in a Gauss-Seidel manner, i.e., $Q_\lambda(\Theta)$ 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 (2.3) is given by Algorithm 1.

Algorithm 1: Coordinate Gradient Descent Algorithm to solve (2.3)

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

repeat

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

$$II^{(k)} = \text{diag}\left[\min\left\{\max\left\{\left[\nabla^2 f(\Theta^{(k)})\right]_{jj}, 10^{-2}\right\} 10^{9}\right\}\right]_{j=1,\ldots,p} \tag{2.4}$$

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

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

$$d_{H^{(k)}}(\Theta_j^{(k)}) \leftarrow \arg\min_{d}\left\{\nabla f(\Theta_j^{(k)})d + \frac{1}{2}d^2 H_{jj}^{(k)} + \lambda P(\Theta_j^{(k)} + d)\right\} \tag{2.5}$$

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

$$Q_\lambda(\Theta_j^{(k)} + \alpha^{(k)} d^{(k)}) \leq Q_\lambda(\Theta_j^{(k)}) + \alpha^{(k)} \varrho \Delta^{(k)} \tag{2.6}$$

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

$$\Delta^{(k)} := \nabla f(\Theta_j^{(k)}) d^{(k)} + \gamma (d^{(k)})^2 H_{jj}^{(k)} - \lambda P(\Theta_j^{(k)} + d^{(k)}) - \lambda P(\Theta_j^{(k)}) \tag{2.7}$$

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 (2.2). Without loss of generality, we assume the penalty factors (v_j) are all equal to 1.

In all chapters of this thesis?

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

$$d_H(\Theta_j) = \arg\min_d G(d) \tag{2.8}$$

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 $0 \in \partial G(d)$:

$$\partial G(d) = \nabla f(\Theta_j) + d H_{jj} + \lambda u \tag{2.9}$$