The Cluster Elastic Net for High-Dimensional Regression With Unknown Variable Grouping

Daniela M. Witten, Ali Shojaie, Fan Zhang

May 17, 2013

Abstract

In the high-dimensional regression setting, the elastic net produces a parsimonious model by shrinking all coefficients towards the origin. However, in certain settings, this behavior might not be desirable: if some features are highly correlated with each other and associated with the response, then we might wish to perform less shrinkage on the coefficients corresponding to that subset of features. We propose the *cluster elastic net*, which selectively shrinks the coefficients for such variables towards each other, rather than towards the origin. Instead of assuming that the clusters are known *a priori*, the cluster elastic net infers clusters of features from the data, on the basis of correlation among the variables as well as association with the response. These clusters are then used in order to more accurately perform regression. We demonstrate the theoretical advantages of our proposed approach, and explore its performance in a simulation study, and in an application to HIV drug resistance data. Supplementary Materials are available online.

Keywords: correlated variables, feature selection, feature clustering, structured sparsity, lasso, ridge, $p \gg n$

1 Introduction

In this paper, we consider the problem of performing linear regression with a response vector $\mathbf{y}$ of length n and a data matrix $\mathbf{X}$ of dimension $n \times p$, where p is the number of features and n is the number of observations. Least squares linear regression involves estimating the coefficient vector $\boldsymbol{\beta}$ by minimizing the sum of squared errors $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$. Unfortunately, it cannot be performed when $\mathbf{X}$ is singular, for instance in high dimensions when $p > n$.

In recent years, a great number of proposals have been made to overcome this limitation of least squares regression. *Ridge regression* involves selecting the coefficient vector $\boldsymbol{\beta}$ that minimizes the sum of squared errors, subject to a squared ℓ_2 penalty (Hoerl & Kennard 1970). Unfortunately, ridge regression does not produce parsimonious models — the resulting coefficient estimate $\hat{\boldsymbol{\beta}}$ contains no elements that are exactly equal to zero. In contrast, the *lasso* (Tibshirani 1996) achieves sparse coefficient estimates by minimizing the sum of squared

errors with an ℓ_1 penalty on the coefficient vector. But the lasso has a major shortcoming relative to ridge regression: while ridge regression tends to assign similar coefficient values to correlated variables, the lasso tends to only assign a non-zero coefficient to a single variable out of a set of correlated variables. To combine ridge regression's treatment of correlated variables with the lasso's sparsity, Zou & Hastie (2005) proposed the *elastic net*, which combines an ℓ_1 and a squared ℓ_2 penalty on $\boldsymbol{\beta}$, and achieves model parsimony along with a tendency for correlated variables to yield similar regression coefficients.

In certain settings, it may be known *a priori* that there are distinct *groups* or *clusters* among the variables, and we may wish to exploit these groups when performing regression. For instance, we may wish to predict some response $\mathbf{y}$ on the basis of a data matrix $\mathbf{X}$ consisting of p gene expression measurements for n observations. It is known that genes operate as part of *pathways*. If the pathways are known, then we could encourage the variables within a group to have a shared pattern of sparsity — that is, to all be zero or to all be non-zero. The *group lasso* proposal of Yuan & Lin (2007) achieves this, through the use of an ℓ_2 penalty on the coefficients within each of K known and non-overlapping groups. Some modifications to this proposal have been made to allow for an additional lasso penalty to encourage sparsity for individual elements within a group (Simon et al. 2010), to accommodate overlapping groups (Jacob et al. 2009), and to encourage a shared sign for the non-zero coefficients within each group (Chiquet et al. 2012). Another setting in which it may be beneficial to encourage similarity in estimated coefficients is when a known graph structure for the covariates is available. In this case, we can perform *graph-constrained* regression: this is achieved by performing regression subject to a penalty that encourages covariates that are linked on the graph to take on similar coefficients (Li & Li 2008, Li & Li 2010, Huang et al. 2011, Shen et al. 2012).

The group lasso and graph-constrained regression proposals just described can be used to exploit external information about the covariates in order to potentially obtain more accurate results in high-dimensional settings. However, what if no such external information is available? For instance, in genetic studies, a given set of pathways may not be relevant to a response of interest, and so using these known pathways may not lead to improved results. In such a setting, rather than using known groups in order to exploit covariate structure in

regression, we might want to estimate the groups, or *clusters*, from the data. In this paper, we propose the *cluster elastic net* (CEN), an approach for identifying clusters among the variables and simultaneously estimating the regression coefficients. We will show that in the absence of clusters our proposal is equivalent to the elastic net (Zou & Hastie 2005), and in the presence of known clusters is closely related to graph-constrained regression (Li & Li 2008, Li & Li 2010). But in the presence of unknown clusters — which is the case in general, and is the scenario of interest in this paper — our approach is novel and outperforms existing approaches by encouraging features within a cluster to have a shared association with the response.

We are not the first to propose performing clustering together with regression. Several algorithmic proposals have been made for performing clustering and then subsequently performing regression using the cluster outputs (Hastie et al. 2001, Dettling & Buhlmann 2004, Park et al. 2007). Penalized regression approaches have also been proposed for exploiting correlation among features in order to obtain improved regression coefficient estimates. For instance, *octagonal shrinkage and clustering algorithm for regression* (OSCAR, Bondell & Reich 2008) and the more recent *penalized adaptive clustering and sparsity* (PACS, Sharma et al. 2013) approaches encourage correlated variables to take on identical coefficient estimates via the use of a novel penalty function that can be interpreted as an octagonal constraint region. Related approaches are proposed in She (2010), Daye & Jeng (2009), and Tutz & Ulbricht (2009). However, while these proposals encourage correlated features to take on the same or similar coefficient values, they do not explicitly encourage *large sets of correlated features* to take on similar coefficient values. More closely related to our proposal is recent work by Buhlmann et al. (2012) on the *cluster group lasso*. This approach involves first identifying groups among the features using (for instance) hierarchical clustering, and then applying the group lasso of Yuan & Lin (2007) to the resulting groups. However, this technique assumes that all correlated features have similar associations with the response. In contrast, CEN seeks sets of correlated features with similar associations with the response; this is particularly advantageous if some but not all correlated features have a similar association with the response.

We now illustrate the performance of CEN in a toy example. We generate an $n \times p$ matrix

$\mathbf{X}$ with $n = 50$ and $p = 30$. The rows of $\mathbf{X}$ are i.i.d. draws from a $N(0, \mathbf{\Sigma})$ distribution, where $\mathbf{\Sigma}$ is a $p \times p$ block diagonal matrix with three equally-sized blocks. $\mathbf{\Sigma}$ has 1's on the diagonal, 0.8's within each block, and 0's elsewhere. Figure 1(a) is a heatmap illustrating the empirical correlation matrix of $\mathbf{X}$. The coefficient vector $\boldsymbol{\beta}$ takes the form

$$\boldsymbol{\beta} = \left(\underbrace{1, \ldots, 1}_{10}, \underbrace{-1, \ldots, -1}_{10}, \underbrace{0, \ldots, 0}_{10} \right)^T .$$

Finally, we generate the response according to $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$, where the elements of $\boldsymbol{\epsilon}$ are i.i.d. draws from a $N(0, 1)$ distribution. We then set the tuning parameter for the ℓ_1 penalty term in CEN to zero. Figures 1(b)-(d) show the $\boldsymbol{\beta}$ estimates that result from performing CEN, PACS (discussed in Section 5.2, Sharma et al. 2013), and ridge regression, with tuning parameters chosen so that the resulting estimators have the same ℓ_2 norms. We see that in this example, CEN yields the most compact and accurate coefficient estimates within each cluster. Figures 1(e)-(h) extend this example to the setting in which each of the three groups contains both positively-correlated and negatively-correlated features, as shown in Figure 1(e). The coefficient vector is such that $X_j\beta_j \approx X_l\beta_l$ for the features within each group. CEN performs very well in this situation, indicating that this approach can handle both negative and positive correlations among the features within a group, provided that the features share an association with the response.

The rest of this paper is organized as follows. In Section 2, we present the CEN optimization problem and its properties. An algorithm for solving the optimization problem is given in Section 3. In Section 4, we investigate the differences between the shrinkage performed by CEN and that performed by the elastic net. We study CEN's relationship with other approaches in the literature in Section 5. In Section 6, we study CEN's performance in a simulation study. An application to HIV drug resistance data is presented in Section 7, and the discussion is in Section 8.

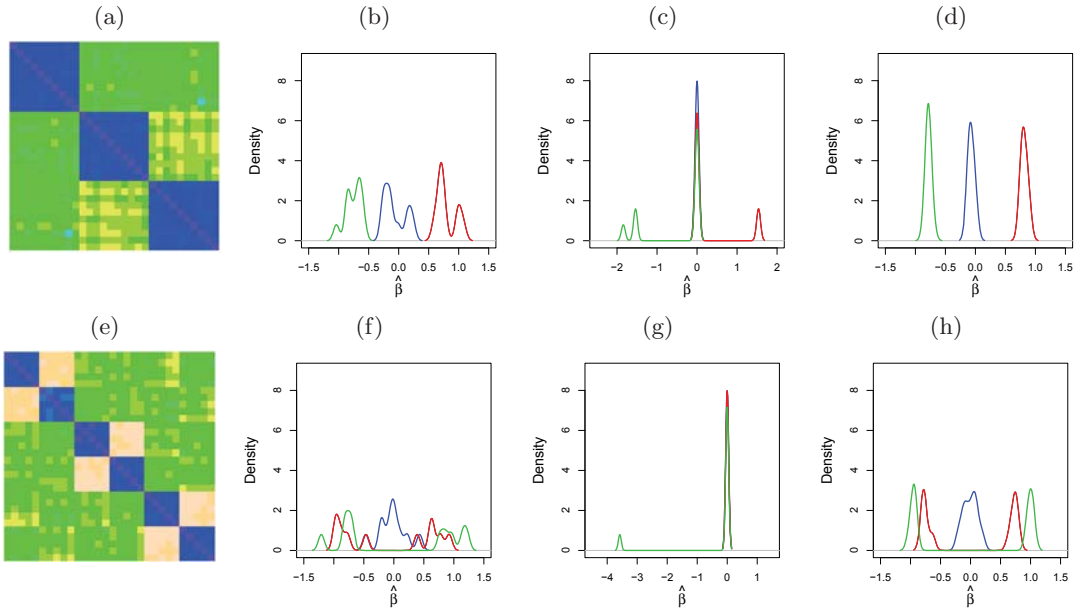

Figure 1: Grouping properties of CEN with $p = 30$ and $n = 50$. *(a)-(d):* There are three highly correlated sets of features, with coefficient values 1, 0, and -1, shown in green, blue, and red, respectively. *(a):* Heatmap of empirical correlation of the design matrix $\mathbf{X}$; positive, zero, and negative correlations are indicated in blue, green, and beige. *(b):* Density plot of $\hat{\boldsymbol{\beta}}$ using ridge regression. *(c):* Density plot of $\hat{\boldsymbol{\beta}}$ using PACS. *(d):* Density plot of $\hat{\boldsymbol{\beta}}$ using CEN with $\delta = 0$. *(e)-(h):* As in panels (a)-(d), but now each group contains both positively- and negatively-correlated features. In the blue group all coefficients equal zero. Within the red and green groups, half the coefficients equal 1 and half equal -1, such that $X_j\beta_j \approx X_l\beta_l$ for all features within each group. This is captured by CEN (h) but not by ridge (f) or PACS (g).

2 The Cluster Elastic Net

2.1 The CEN Optimization Problem

Throughout this paper, we will assume a fixed design matrix $\mathbf{X}$ of dimension $n \times p$ and a response vector $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta}$ is an unknown vector of regression coefficients and $\boldsymbol{\epsilon}$ is a random vector of uncorrelated noise terms with mean 0 and common variance σ^2. Furthermore, we assume that $\mathbf{y}$ has been centered to have mean zero. We will let $\mathbf{X}_j \in \mathbb{R}^n$ denote the jth column of the matrix $\mathbf{X}$. We will also assume that the columns of $\mathbf{X}$ have been standardized to have mean 0 and an ℓ_2 norm of 1: $\sum_i X_{ij} = 0, \sum_i X_{ij}^2 = 1$. In general, we will assume that we are in the high-dimensional, sparse setting in which $p > n$, but the majority of the covariates are not associated with the outcome, i.e. $\beta_j = 0$ for most $j = 1, \ldots, p$. We make the following additional assumptions:

Assumption 1. There are unknown groups, or *clusters*, among the variables. *There are K distinct but unknown groups of variables, with moderate or high levels of (absolute) correlation among the variables within a group, and little or no correlation between the groups.*

Assumption 2. Variables that are in the same group have a similar association with the response. *If $\mathbf{X}_j$ and $\mathbf{X}_l$ belong to the same group, then $\mathbf{X}_j\beta_j$ and $\mathbf{X}_l\beta_l$ take on similar values.*

The last two assumptions indicate that there are unknown groups among the variables, and that knowing these groups would allow us to more accurately estimate $\boldsymbol{\beta}$. With these assumptions in mind, we propose the *cluster elastic net*, which is the solution to the following optimization problem:

$$\underset{C_1,\ldots,C_K,\boldsymbol{\beta}}{\text{minimize}} \left\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \delta\|\boldsymbol{\beta}\|_1 + \frac{\lambda}{2}\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{j,l\in C_k}\|\mathbf{X}_j\beta_j - \mathbf{X}_l\beta_l\|^2 \right\}. \tag{1}$$

Here δ and λ are nonnegative tuning parameters, and $C_1,\ldots,C_K$ denotes a partition of the p features into K groups, such that $C_k \cap C_l = \emptyset$ if $k \neq l$ and $C_1 \cup C_2 \cup \ldots \cup C_K = \{1,\ldots,p\}$. The $\|\boldsymbol{\beta}\|_1$ term is simply a lasso (ℓ_1) penalty, which will encourage the coefficient estimates to be sparse when δ is large. On the other hand, the behavior of the *cluster penalty*, which can also be re-written as follows,

$$\frac{1}{2}\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{j,l\in C_k}\|\mathbf{X}_j\beta_j - \mathbf{X}_l\beta_l\|^2 = \sum_{k=1}^{K}\sum_{j\in C_k}\|\mathbf{X}_j\beta_j - \frac{1}{|C_k|}\sum_{l\in C_k}\mathbf{X}_l\beta_l\|^2, \tag{2}$$

is more subtle. Assume for a moment that the clusters $C_1,\ldots,C_K$ are known. Then when λ is large, the cluster penalty term will encourage $\mathbf{X}_j\beta_j \approx \mathbf{X}_l\beta_l$ for $j,l \in C_k$.

In this paper, we are interested in the setting where the clusters are unknown, and so the cluster penalty term will encourage these clusters to be selected on the basis of the $\mathbf{X}_j\beta_j$'s. As we will see shortly, we can solve the optimization problem (1) by repeatedly estimating the clusters $C_1,\ldots,C_K$ by performing k-means clustering of the $\mathbf{X}_j\beta_j$'s, and then estimating $\boldsymbol{\beta}$ by encouraging variables within a given cluster to take on similar coefficient estimates. If the jth and lth features are in the same cluster and have high (absolute) correlation, then (2) encourages β_j and β_l to take on similar values. If the jth and lth features are in the same cluster and have low correlation, then the penalty encourages β_j and β_l to be near zero.

In what follows, we will refer to (1) as the *CEN optimization problem*. Occasionally, we will also consider a simpler version of (1) with $C_1, \ldots, C_K$ fixed — we will refer to this modified version of CEN as *CEN with known clusters*. We will also refer to the special case of CEN when $\delta = 0$ as *cluster ridge regression* (CRR) due to similarities between the ℓ_2 penalty and the cluster penalty, which we will explore shortly.

2.2 Properties of the Cluster Elastic Net

We first show that lasso and elastic net are special cases of CEN when $K = 1$ or $K = p$.

Property 1. *If $K = p$, so that each feature is in its own cluster, then CEN reduces* exactly *to the lasso.*

Property 2. *If $K = 1$, so that all features are in the same cluster, then CEN is equivalent to the elastic net on scaled versions of $\mathbf{X}$ and $\mathbf{y}$.*

Property 1 can be seen by inspection of (1), but Property 2 requires further comment. Note that when $K = 1$, then provided that $\lambda < p$, we can use (2) to rewrite (1) as

$$\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\beta}\|^2 + \delta\|\boldsymbol{\beta}\|_1 + \lambda\|\boldsymbol{\beta}\|^2,$$

where $\tilde{\mathbf{y}} = \mathbf{y}/\sqrt{1-\lambda/p}$ and $\tilde{\mathbf{X}} = \mathbf{X}\sqrt{1-\lambda/p}$, and where we are omitting an additive constant that is a function of only $\mathbf{y}$, λ, and p. Together, Properties 1 and 2 indicate that the CEN defines a spectrum of regularized regression problems, at one end of which is the lasso ($K = p$) and at the other end of which is the elastic net ($K = 1$).

We will next show that for an intermediate value of K, $1 < K < p$, CEN will result in pooling of regression coefficients for variables within a cluster, provided that those variables are correlated. Letting $r_{jl} \equiv \mathbf{X}_j^T\mathbf{X}_l$, we can re-write the objective of (1) as follows:

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \frac{\lambda}{2}\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{j,l\in C_k}\left[(1-r_{jl})\left(\beta_j^2+\beta_l^2\right) + r_{jl}\left(\beta_j-\beta_l\right)^2\right] + \delta\left\|\boldsymbol{\beta}\right\|_1. \tag{3}$$

Therefore, if the jth and lth variables are in the same group and r_{jl} is large, then the $(\beta_j - \beta_l)^2$ term in (3) will dominate and CEN will shrink β_j and β_l *towards each other*. On the other

hand, if r_{jl} is close to zero, then $(\beta_j^2 + \beta_l^2)$ — a term that amounts to a ridge penalty on a subset of the variables — will dominate and CEN will shrink β_j and β_l towards zero. And if r_{jl} is negative, then

$$(1 - r_{jl})\left(\beta_j^2 + \beta_l^2\right) + r_{jl}\left(\beta_j - \beta_l\right)^2 = (1 - |r_{jl}|)\left(\beta_j^2 + \beta_l^2\right) + |r_{jl}|\left(\beta_j + \beta_l\right)^2,$$

which indicates that $\beta_j \approx -\beta_l$ is encouraged. In other words, depending on the correlations among the variables within a group, the variables will either be shrunken *towards each other*, *towards zero*, or *towards each other in absolute value but with opposite signs.*

We also observe from (1) that (absolutely) correlated variables that are associated with the response — that is, variables for which $X_j\beta_j \approx X_k\beta_k$ — are encouraged to belong to the same cluster, as this results in less shrinkage in their coefficients, and hence, smaller values of the objective.

Furthermore, we note that

$$\frac{1}{2}\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{j,l\in C_k}\left(\beta_j^2 + \beta_l^2 - 2\beta_j\beta_l r_{jl}\right) = \beta^T\mathbf{M}\beta \tag{4}$$

where $\mathbf{M}$ is a positive semi-definite matrix of the form

$$M_{jl} = \begin{cases} (|C_k| - 1)/|C_k| & \text{if } j = l \in C_k \\ -r_{jl}/|C_k| & \text{if } j \neq l \text{ and } j, l \in C_k \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

and M_{jl} is the (j, l) entry of $\mathbf{M}$. Therefore, the optimization problem for CEN with known groups can equivalently be written as

$$\underset{\boldsymbol{\beta}}{\text{minimize}}\left\{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\boldsymbol{\beta}^T\mathbf{M}\boldsymbol{\beta} + \delta\|\boldsymbol{\beta}\|_1\right\}, \tag{6}$$

where $\mathbf{M}$ is a matrix that effectively *reduces* the amount of penalization that is applied to pairs of correlated variables within a given group. If $\mathbf{M} \propto \mathbf{I}$, as will be the case if the design matrix has orthogonal columns and $|C_1| = |C_2| = \ldots = |C_K|$, then (6) reduces to the elastic

net. This results in the following property.

Property 3. *In the case of an orthogonal design matrix,* $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$*, and equally-sized clusters,* $|C_1| = |C_2| = \ldots = |C_K|$*, the CEN is equivalent to the elastic net.*

Therefore, when $K = p$, $K = 1$, or $\mathbf{X}$ has orthogonal columns, the CEN reduces to the lasso or the elastic net. When $K = 1$, (3) also reveals a very close connection between CEN and the proposal of Tutz & Ulbricht (2009). However, when $1 < K < p$ and the columns of $\mathbf{X}$ are not orthogonal, the CEN yields a new regularization procedure. Unlike the elastic net or the lasso, it shrinks coefficients towards each other or towards the origin based on the pairwise correlations of features that belong to the same cluster.

2.3 Coefficient Profiles

Figure 2 displays the coefficient profiles of five regression techniques — the elastic net, the group lasso, PACS, CEN, and CEN with known groups — on the toy example from Figures 1(a)-(d). We see that the coefficients corresponding to features within a given cluster are grouped together far more tightly by CEN than by the elastic net. Furthermore, the group lasso (which can only be performed if the clusters are known) yields grossly inaccurate coefficient estimates because it does not encourage similar coefficient estimates for features within a cluster. PACS fails to group together the coefficient estimates within each cluster: only a few truly relevant variables are given non-zero coefficient values before the truly irrelevant variables. We also see that in this setting, the results of CEN with unknown clusters and CEN with clusters known *a priori* are virtually indistinguishable.

2.4 Contour Plots for CEN

In order to better understand the CEN penalty relative to existing penalties, we consider its contour plots. Figure 3 displays the contour plots of the penalty function $P(\boldsymbol{\beta})$, where $P(\boldsymbol{\beta})$ is a lasso penalty (Figure 3(a)), ridge penalty (Figure 3(b)), elastic net penalty (Figure 3(c)), or a CEN penalty (Figures 3(d)-(f)). In particular, Figures 3(d)-(f) display the contours of the CEN penalty under the following three scenarios:

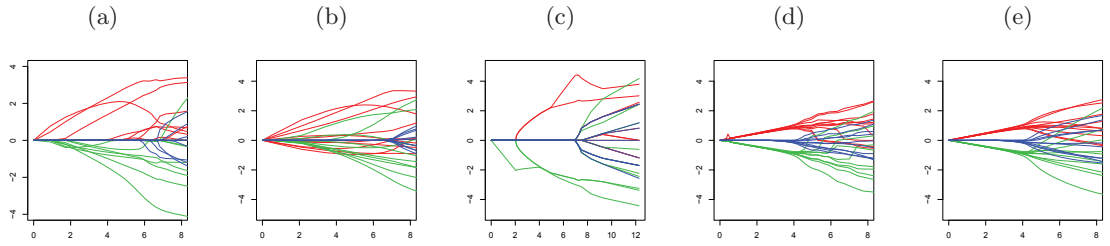

Figure 2: Coefficient profiles for *(a):* elastic net, *(b):* group lasso, *(c):* PACS, *(d):* CEN, and *(e):* CEN with known groups. The ℓ_2 norm of the estimated coefficient vector is displayed on the x-axis, and the coefficient estimates are on the y-axis. Colors indicate the cluster to which each set of features belongs; red, green, and blue indicate true coefficient values of 1, -1, and 0, respectively.

Figure 3(d): Here there are $p = 2$ positively-correlated features and $K = 1$, so that

$$P(\boldsymbol{\beta}) = (\lambda/2)\|\mathbf{X}_1\beta_1 - \mathbf{X}_2\beta_2\|^2 + \delta|\beta_1| + \delta|\beta_2| = (\lambda/2)(\beta_1^2 + \beta_2^2 - 2\beta_1\beta_2 r_{12}) + \delta|\beta_1| + \delta|\beta_2|.$$

Therefore, the contours for $P(\boldsymbol{\beta})$ are an ellipse (centered at the origin) plus a diamond.

Figure 3(e): Here there are $p = 4$ positively-correlated predictors belonging to a single cluster. β_1 is on the x-axis, and β_2 is on the y-axis. We assume that $\beta_3 = \beta_4 = 1$. Contour plots for $P(\boldsymbol{\beta})$ are shown, with β_3 and β_4 held fixed at their true values. In other words, contour plots for

$$(\lambda/4)\left(||\mathbf{X}_1\beta_1 - \mathbf{X}_2\beta_2||^2 + \sum_{j=3}^{4}\left(||\mathbf{X}_1\beta_1 - \mathbf{X}_j||^2 + ||\mathbf{X}_2\beta_2 - \mathbf{X}_j||^2\right)\right) + \delta|\beta_1| + \delta|\beta_2|$$

are displayed. This is the sum of an ellipse (not centered at the origin) and a diamond. This indicates that β_1 and β_2 are encouraged to take on similar positive values.

Figure 3(f): Here there are $p = 8$ predictors and $K = 2$. β_1 is on the x-axis, and β_5 is on the y-axis. The first four features are highly correlated and belong to a cluster, as do the remaining four features. Furthermore, we assume that $\beta_2 = \beta_3 = \beta_4 = 1$ and that $\beta_6 = \beta_7 = \beta_8 = -1$. Contour plots for $P(\boldsymbol{\beta})$, with $\beta_2, \beta_3, \beta_4, \beta_6, \beta_7, \beta_8$ held fixed at

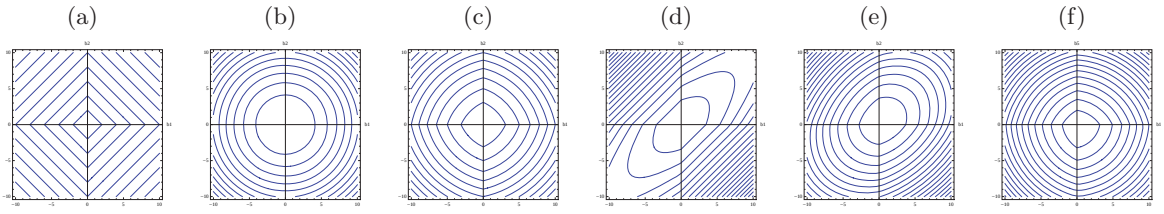

Figure 3: Contour plots are shown for *(a)* the lasso; *(b)* ridge regression; *(c)* the elastic net; *(d)* CEN with $p = 2$, positive correlation between the variables, and $K = 1$; *(e)* CEN with two positively-correlated variables in the same cluster; *(f)* CEN with two uncorrelated variables in different clusters. Details about *(d)*-*(f)* are provided in the text.

their true values, are shown. That is, we display

$$(\lambda/8)\left(\sum_{j=2}^{4}\|\mathbf{X}_1\beta_1 - \mathbf{X}_j\|^2 + \sum_{j=6}^{8}\|\mathbf{X}_5\beta_5 + \mathbf{X}_j\|^2\right) + \delta|\beta_1| + \delta|\beta_5|;$$

this is the sum of a circle (not centered at the origin) and a diamond. Consequently, β_1 and β_5 are encouraged to take on positive and negative values, respectively.

Therefore, we see that unlike ridge regression, the lasso, and the elastic net, the contours of CEN are *data driven*: their shape depends on the correlation structure of the design matrix, $\mathbf{X}$. If the design matrix is orthogonal, then the contours will simply be circles (centered at the origin) plus diamonds; this amounts to the contours of the elastic net. In contrast, in the presence of high correlation between features within a cluster, the contours corresponding to features within a cluster will be ellipses (not centered at the origin) plus diamonds, and the contours corresponding to pairs of features in different clusters will be circles (not centered at the origin) plus diamonds. In the presence of a non-orthogonal design matrix, the contours of the penalty function are not centered at the origin because correlated features are encouraged to take on similar coefficient values.

3 Computational Considerations

3.1 Algorithm for the CEN Problem

We now consider the task of solving the CEN problem (1). This problem is non-convex, and finding the global optimum would require considering all $O(K^p)$ possible partitions of the p

features into K clusters. Then for each partition, (1) could be solved with $C_1, \ldots, C_K$ held fixed. Unfortunately, this exhaustive approach is infeasible unless p is very small.

Therefore, instead of seeking the global optimum to (1), we seek a local optimum. In particular, we take an iterative approach, in which we hold $C_1, \ldots, C_K$ fixed and solve with respect to $\boldsymbol{\beta}$, and then hold $\boldsymbol{\beta}$ fixed and solve with respect to $C_1, \ldots, C_K$. In the latter step, we find a local optimum of (1) with $\boldsymbol{\beta}$ held fixed by applying the k-means clustering algorithm on $X_1\beta_1, \ldots, X_p\beta_p$ (see e.g. Hastie et al. 2009). Details are presented in Algorithm 1.

Since the CEN optimization problem is not convex, Algorithm 1 is not guaranteed to converge to the global optimum. However, it is a descent algorithm in which each iteration decreases the objective. Since we initialize the algorithm using the elastic net coefficient estimates, and since each iteration reduces the objective, the algorithm yields quite good empirical results that improve upon the elastic net by exploiting grouping among the variables.

We now consider solving (9) in Step 2(b) of Algorithm 1. The problem is convex in $\boldsymbol{\beta}$. We take a coordinate descent approach (see e.g. Friedman et al. 2007), which amounts to repeatedly performing a single update, given in Proposition 1.

Proposition 1. *Let $\mathbf{X}_{-j}$ be the $n \times (p-1)$ submatrix containing all but the jth column of $\mathbf{X}$, $\boldsymbol{\beta}_{-j}$ the $(p-1)$-vector containing all but the jth element of $\boldsymbol{\beta}$, $\tilde{\mathbf{y}}_j = \mathbf{y} - \mathbf{X}_{-j}\boldsymbol{\beta}_{-j}$, and suppose that $j \in C_k$. Then the following update to β_j minimizes the objective function in (9) with respect to β_j while holding all other variables fixed:*

$$\beta_j \leftarrow \frac{S\left(\tilde{\mathbf{y}}_j^T \mathbf{X}_j + \frac{\lambda}{|C_k|}\sum_{l \in C_k, j \neq l} \beta_l r_{jl}, \delta/2\right)}{r_{jj}\left(1 + \lambda\frac{|C_k|-1}{|C_k|}\right)}. \tag{7}$$

where S indicates the soft-thresholding operator, defined as $S(a,b) = \text{sign}(a)\max(0, |a| - b)$.

Therefore, to solve (9), we simply iterate through the variables $j = 1, \ldots, p$, repeating the update (7) until convergence to the global optimum. We see from the form of (7) that β_j will be encouraged to take on large values if correlated variables that are in the same cluster also take on large values. The proof of Proposition 1 follows from simple algebra and is omitted.

Algorithm 1 Algorithm for solving the CEN optimization problem (1)

1. Initialize $\boldsymbol{\beta}$ as the solution to the elastic net optimization problem,

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \left\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \delta\|\boldsymbol{\beta}\|_1 + \lambda\|\boldsymbol{\beta}\|^2 \right\}.$$

2. Iterate until convergence:

   (a) Hold $\boldsymbol{\beta}$ fixed and minimize (1) with respect to $C_1, \ldots, C_K$. That is, solve

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{j,l \in C_k} \|\mathbf{X}_j\beta_j - \mathbf{X}_l\beta_l\|^2 \right\}. \tag{8}$$

   A local optimum can be found by performing k-means clustering on $\mathbf{X}_1\beta_1, \ldots, \mathbf{X}_p\beta_p$ with K clusters.

   (b) Hold $C_1, \ldots, C_K$ fixed and and solve for $\boldsymbol{\beta}$. That is, solve

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \left\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \frac{\lambda}{2} \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{j,l \in C_k} \|\mathbf{X}_j\beta_j - \mathbf{X}_l\beta_l\|^2 + \delta\|\boldsymbol{\beta}\|_1 \right\}. \tag{9}$$

3.2 Computations

In our implementation of Algorithm 1, k-means clustering is performed in Step 2(a) using the `kmeans` function with `nstart=20` in the `MASS` library in `R`. Coordinate descent is performed in Step 2(b) using our own implementation in `R`.

In principle, each time Step 2(b) is performed, the computations should be comparable to performing the elastic net once using the coordinate descent approach of Friedman et al. (2007). However, to speed up computations in our implementation, instead of iterating the update (7) until the global optimum is obtained in Step 2(b), we perform at most 50 iterations of the update (7) each time Step 2(b) is performed. We iterate between Steps 2(a) and 2(b) until the relative change in the estimated coefficients, $\|\hat{\boldsymbol{\beta}}^{(i)} - \hat{\boldsymbol{\beta}}^{(i-1)}\|^2 / \|\hat{\boldsymbol{\beta}}^{(i)}\|^2$, falls below 10^{-5}, where $\hat{\boldsymbol{\beta}}^{(i)}$ denotes the coefficient estimates from the ith iteration of Step 2.

On a MacBook Pro 2.66 GHz Intel Core i7, running CEN in the simulation set-up of Section 6 (for which $n = 200$ and $p = 1000$) took an average of 5 seconds. Computations can be reduced using warm starts over a grid of λ or δ values, or using an active set approach.

3.3 Tuning Parameter Selection

In order to select the tuning parameters δ, λ, and K, cross-validation or a training set / test set approach can be used. Recall from Section 2.2 that for certain values of λ and K, CEN simplifies to the lasso or the elastic net. Therefore, if for a particular data set the assumptions underlying CEN do not hold, cross-validation should in principle result in a selection of tuning parameters such that either the lasso or the elastic net is performed. Furthermore, Properties 1 and 2 suggest that a broad range of K values might give good results.

4 Analysis of Between-Group Shrinkage

While the elastic net shrinks all coefficient estimates towards the origin, the cluster elastic net shrinks the coefficients for correlated features that belong to the same cluster towards each other instead of towards the origin. We explore this property in a very simple setting, in which (for simplicity) we take the clusters to be known. We make the following assumptions:

(A1) There are two known clusters, each with size $m = p/2$. The features are ordered such that those in the first cluster precede those in the second cluster.

(A2) The true $\boldsymbol{\beta}$ is $(b_1, \ldots, b_1, b_2, \ldots, b_2)^T$. That is, the true value of $\boldsymbol{\beta}$ is the same within each cluster. We also assume without loss of generality that $b_1 > b_2$.

(A3) $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_n)$.

(A4) $r_{jl} = r_1$ for j and l in the same cluster, and $r_{jl} = r_0$ for j and l in different clusters. This means that

$$\mathbf{X}^T\mathbf{X} = (1 - r_1)\,\mathbf{I} + \begin{bmatrix} r_1 & \cdots & r_1 & r_0 & \cdots & r_0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ r_1 & \cdots & r_1 & r_0 & \cdots & r_0 \\ r_0 & \cdots & r_0 & r_1 & \cdots & r_1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ r_0 & \cdots & r_0 & r_1 & \cdots & r_1 \end{bmatrix}.$$

Furthermore, we assume that $r_1 > r_0 > 0$.

While assumptions (A1)-(A3) are reasonable, assumption (A4) is quite simplistic. Note that with a random design matrix, (A4) corresponds to a simple block-correlation model, and thus is reasonable for observed data as $n \to \infty$.

Now, we will compare the CRR estimator $\hat{\boldsymbol{\beta}}_{\text{CRR}}$ to the ridge regression estimator $\hat{\boldsymbol{\beta}}_{\text{RR}}$. It is straightforward to show that both estimators are normally distributed.

Theorem 1. *Suppose that (A1)-(A4) hold, and that* $\lambda_{\text{RR}} = \left(1 - \frac{1-r_1}{m}\right) \lambda_{\text{CRR}}$.

1. *If the* j*th and* l*th features are in the same cluster, then*

$$\hat{\beta}_{\text{CRR},j} - \hat{\beta}_{\text{CRR},l} \overset{D}{=} \hat{\beta}_{\text{RR},j} - \hat{\beta}_{\text{RR},l} \sim N\left(0, 2\sigma^2 \left(1 + \lambda_{\text{RR}} - r_1\right)^{-2} (1 - r_1)\right).$$

2. *If the* j*th and* l*th features are in different clusters, then*

$$E(\hat{\beta}_{\text{CRR},j} - \hat{\beta}_{\text{CRR},l}) = \frac{(\beta_j - \beta_l)(-1 + m(r_0 - r_1) + r_1)}{\lambda_{\text{CRR}}(-1+m)(-1+r_1)/m + (-1 + m(r_0 - r_1) + r_1)}, \quad (10)$$

$$E(\hat{\beta}_{\text{RR},j} - \hat{\beta}_{\text{RR},l}) = \frac{(\beta_j - \beta_l)(-1 + m(r_0 - r_1) + r_1)}{-1 - \lambda_{\text{RR}} + m(r_0 - r_1) + r_1}. \quad (11)$$

Corollary 1. *Under the assumptions of Theorem 1, if the* j*th and* l*th features are in different clusters, then*

$$1 \geq \frac{E(\hat{\beta}_{\text{CRR},j} - \hat{\beta}_{\text{CRR},l})}{\beta_j - \beta_l} \geq \frac{E(\hat{\beta}_{\text{RR},j} - \hat{\beta}_{\text{RR},l})}{\beta_j - \beta_l} \geq 0.$$

Furthermore, if $r_1 = 1$*, then*

$$\frac{E(\hat{\beta}_{\text{CRR},j} - \hat{\beta}_{\text{CRR},l})}{\beta_j - \beta_l} = 1 \quad \textit{and} \quad \frac{E(\hat{\beta}_{\text{RR},j} - \hat{\beta}_{\text{RR},l})}{\beta_j - \beta_l} = \frac{m(1 - r_0)}{\lambda_{\text{RR}} + m(1 - r_0)} < 1.$$

Theorem 1 (proven in the Supplementary Materials) and Corollary 1 can be interpreted as follows. Theorem 1 indicates that there exists a simple relationship between λ_{RR} and λ_{CRR} such that $\hat{\beta}_{\text{CRR},j} - \hat{\beta}_{\text{CRR},l}$ and $\hat{\beta}_{\text{RR},j} - \hat{\beta}_{\text{RR},l}$ have the same distribution, provided that the jth and lth features are in the same cluster. In other words, with tuning parameters chosen in this way, CRR and RR perform *the same amount of shrinkage within a cluster*.

However, how does the shrinkage performed by CRR and RR compare for features in *different clusters*? Corollary 1 reveals that the tuning parameter relationship that leads to

the same amount of within-cluster shrinkage results in *more* between-cluster shrinkage by RR than by CRR. Stated in another way, both RR and CRR successfully shrink coefficients of features within the same cluster towards each other; RR also shrinks together the coefficients of features that are in different clusters. For instance, when $r_1 = 1$ and the jth and lth features are in different clusters, then $E(\hat{\beta}_{\text{CRR},j} - \hat{\beta}_{\text{CRR},l}) = \beta_j - \beta_l$, but $E(\hat{\beta}_{\text{RR},j} - \hat{\beta}_{\text{RR},l}) < \beta_j - \beta_l$. The tendency of RR to perform between-cluster shrinkage is illustrated in Figure 4; as can be seen from the figure, CRR does not exhibit this behavior to the same extent.

Interestingly, Theorem 1 reveals that provided that $r_1 > 0$, then even if $r_0 = 0$ — that is, *even in the absence of any correlation between the features in different clusters* — then RR still shrinks coefficients for features in different clusters towards each other more than does CRR. This is a byproduct of the fact that RR shrinks all coefficients towards zero more than does CRR, regardless of cluster membership. This is an undesirable property of RR.

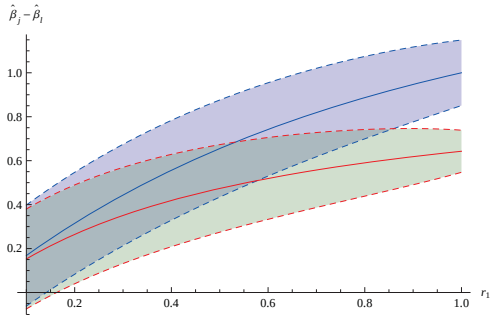

Figure 4: Under the assumptions of Theorem 1, $E(\hat{\beta}_{\text{CRR},j} - \hat{\beta}_{\text{CRR},l})$ (blue solid line) and $E(\hat{\beta}_{\text{RR},j} - \hat{\beta}_{\text{RR},l})$ (red solid line) are displayed as a function of r_1. Here the jth and lth features belong to different groups. Furthermore, $r_0 = 0.1$, $m = 10$, $\beta_j - \beta_l = 1$, and $\lambda_{\text{RR}} = 5$. Unlike CRR, RR tends to shrink together the coefficients of features in different groups. The dashed lines are expectations plus and minus one standard deviation of $\hat{\beta}_j - \hat{\beta}_l$.

5 Relationship With Other Approaches

In Section 2.2, we discussed the relationship of CEN with the lasso and the elastic net. Here we discuss the relationship of CEN with some other recent proposals.

5.1 Relationship With Graph-Constrained Regularization

A series of recent papers have proposed an approach for high-dimensional regression with graph-structured variables (Li & Li 2008, Li & Li 2010). Consider a *weighted graph* $G = (E, V, W)$ where $V = \{1, \ldots, p\}$ is a set of *vertices* that correspond to the p predictors, $E = \{j \sim l\}$ is the set of *edges* between the vertices in the graph, and $w(j, l)$ denotes the (positive) *weight* of the edge between the jth and lth vertices. Let $d_l = \sum_{j \sim l} w(j, l)$ be the *degree* of the lth vertex, and assume that the graph G is known *a priori*. Then the *graph-constrained estimator* (GRACE) of Li & Li (2008) and Li & Li (2010) amounts to solving (6), where $\mathbf{M} = \mathbf{M}^{grace}$, given by

$$M_{jl}^{grace} = \begin{cases} 1 - w(j,j)/d_j & \text{if } j = l \text{ and } d_j \neq 0 \\ -w(j,l)/\sqrt{d_j d_l} & \text{if } j \text{ and } l \text{ share an edge on the graph} \cdot \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Now we will consider a special case of the graph G in order to highlight the relationship between CEN and GRACE. Suppose that the graph G is composed of K disjoint components, and $w(j, l) = 1$ if the jth and lth vertices are in the same component. In other words, within a given component, all vertices are connected and have equal weights. Clearly $w(j, l) = 0$ if the jth and lth vertices are in different components. If we let C_k be a set containing the indices of the vertices in the kth component, for $k = 1, \ldots, K$, then it is easy to see that $d_j = |C_k|$ if the jth vertex is in the kth component. In this case, (12) reduces to

$$M_{jl}^{grace} = \begin{cases} (|C_k| - 1)/|C_k| & \text{if } j = l \in C_k \\ -1/|C_k| & \text{if } j \neq l \text{ and } j, l \in C_k \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Comparing (13) to (5), we find that the $\mathbf{M}$ matrices for CEN and for this special case of GRACE are quite similar. In fact, CEN and this special case of GRACE would be identical if CEN were performed with known groups, and if $r_{jl} = 1$ for $j, l \in C_k$. However, in practice, $|r_{jl}| < 1$ for $j \neq l$. Therefore, the penalty applied by CEN is somewhat milder than the

GRACE penalty. Furthermore, GRACE requires the network to be known *a priori*, whereas in CEN the clusters, and hence the structure of the graph, are inferred from the data.

5.2 Relationship With Pairwise Absolute Clustering and Sparsity

The OSCAR proposal (Bondell & Reich 2008) involves applying an ℓ_∞ penalty to each pair of coefficients. Sharma et al. (2013) showed that OSCAR can be reformulated as

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \left\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \left(\sum_j w_j \beta_j + \sum_{j<k} w_{jk-} |\beta_k - \beta_j| + \sum_{j<k} w_{jk+} |\beta_j + \beta_k| \right) \right\}, \quad (14)$$

where $w_{jk+} = w_{jk-} = \alpha$ for all $1 \leq j < k \leq p$ and where $w_j = 1$ for all $j = 1, \ldots, p$. Sharma et al. (2013) instead consider (14) with $w_{jk+} = (1 + r_{jk})^{-1}, w_{jk-} = (1 - r_{jk})^{-1}$ or $w_{jk+} = 1_{\{r_{jk} < -c\}}, w_{jk-} = 1_{\{r_{jk} > c\}}$ where $1_{\{A\}}$ is an indicator variable that equals one if the event A holds, and equals zero otherwise. This *pairwise absolute clustering and sparsity* (PACS) approach is intended to encourage correlated features to take on similar coefficient values. In contrast, CEN encourages features that share an association with the response to take on similar coefficient values. This distinction can be seen from the form of (3), recalling that $C_1, \ldots, C_K$ are obtained based on $X_1\hat{\beta}_1, \ldots, X_p\hat{\beta}_p$. In both CEN and PACS, the correlation among features plays a role in determining the extent to which features' coefficient estimates are pooled; however, only in CEN does association *with the response* also play a role. Another difference involves the use of an ℓ_1 penalty on pairs of coefficients by PACS (and OSCAR), as opposed to a squared ℓ_2 penalty by CEN; CEN performs a more mild form of shrinkage, as it does not encourage coefficient values to be exactly identical.

The performances of PACS and CEN were compared in Figures 1 and 2.

6 Simulation Study

6.1 Simulation Set-Up

We simulated data according to the model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ with $p = 1,000$ features. The errors $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. from a $N(0, 2.5^2)$ distribution. The observations (rows of $\mathbf{X}$) are i.i.d. from

a $N(0, \mathbf{\Sigma})$ distribution, where $\mathbf{\Sigma}$ is a $p \times p$ block diagonal matrix, with elements as follows:

$$\Sigma_{ij} = \begin{cases} 1 & \text{if } i = j \\ \rho & \text{if } i \leq 50, j \leq 50, i \neq j \\ \rho & \text{if } 51 \leq i \leq 100, 51 \leq j \leq 100, i \neq j \\ 0 & \text{otherwise} \end{cases}. \tag{15}$$

We explored various values of ρ, ranging from 0 to 0.8. Furthermore, $\beta_j \sim \text{Unif}[0.9, 1.1]$ for $1 \leq j \leq 25$, $\beta_j \sim \text{Unif}[-1.1, -0.9]$ for $51 \leq j \leq 75$, and $\beta_j = 0$ otherwise. In other words, there are two sets of 50 correlated features; half of the features in each set are associated with the response. The remaining 900 features are not correlated with each other and not associated with the response. This simulation set-up is motivated by gene pathways, where genes within the same pathway have correlated expression values, but only a fraction of genes in the pathway have expression that is associated with a response of interest.

Using this set-up, we generated a training set of 200 observations, a validation set of 200 observations, and a test set of 800 observations. The training set was used to fit the model, and the validation set was used for purposes of tuning parameter selection only. In greater detail, we fit each approach on the training set using a range of tuning parameter values. We then selected the final model to be the model that yielded the smallest prediction error, defined as $||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}||_2$, on the validation set.

6.2 Simulation Results

We compared the performances of the following approaches: [1] The elastic net (EN). [2] Ridge regression. [3] The lasso. [4] CEN with $K = 3$. [5] CEN with known groups. [6] K-means clustering with $K = 3$, followed by the group lasso on the resulting clusters; this is the *cluster group lasso* (CGL) proposal of Buhlmann et al. (2012). [7] The group lasso (GL) with known groups. Recall that CEN groups variables with similar values of $\mathbf{X}_j\hat{\beta}_j$. Thus, from the perspective of CEN, in this simulation study there are three groups of variables: $C_1 = \{j : 1 \leq j \leq 25\}$, $C_2 = \{j : 51 \leq j \leq 75\}$, and $C_3 = \{1, \ldots, p\} \backslash \{C_1 \cup C_2\}$. We treat these groups as the "true variable clusters" in what follows. We used these clusters in

performing CEN with known groups, and in performing GL with known groups.

In Table 1, we report the following quantities for various values of ρ: [1] The test set prediction error, given as $||\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\hat{\boldsymbol{\beta}}||_2$. [2] Correct sparsity, defined as the fraction of features that are either correctly determined to be zero, or correctly determined to be non-zero: that is, $\frac{1}{p}\sum_{j=1}^{p} A_j$, where

$$A_j = \begin{cases} 1 & \text{if } \hat{\beta}_j = \beta_j = 0 \\ 1 & \text{if } \hat{\beta}_j \neq 0, \beta_j \neq 0 \\ 0 & \text{else} \end{cases}.$$

[3] The number of nonzero elements in $\hat{\boldsymbol{\beta}}$. [4] The Rand Index (Rand 1971), which measures the agreement between the true clusters and estimated clusters. The Rand Index ranges from 0 to 1; a value close to 1 indicates a high level of agreement between the true and estimated clusters, and a value close to 0 indicates a low level of agreement. In the case of CEN and CGL, the estimated clusters are obtained directly via the algorithm, whereas in the case of ridge, lasso, and elastic net, the estimated clusters are obtained by performing k-means clustering on $\mathbf{X}_1\hat{\beta}_1, \ldots, \mathbf{X}_p\hat{\beta}_p$. For CEN with known groups and GL with known groups, the Rand Index necessarily equals one.

As expected, the elastic net always outperforms ridge since $\boldsymbol{\beta}$ is sparse; it also tends to outperform the lasso due to the presence of correlations among the features, especially as ρ increases. The cluster elastic net performs comparably to the elastic net when $0 \leq \rho \leq 0.1$, since in this case there is little or no correlation structure among the features. As the level of correlation within a group increases, the performance of CEN relative to EN improves. When $\rho \geq 0.2$, CEN outperforms EN by a sizeable margin. Not surprisingly, CEN with known groups performs better than CEN with unknown groups, though of course the setting of unknown groups is of primary interest in this paper.

We see that group lasso with known groups performs extremely well across all simulation settings, and far outperforms even CEN with known groups. However, recall that here the known groups used by group lasso are different from the groups used by CGL, because CGL finds groups by clustering $\mathbf{X}_1, \ldots, \mathbf{X}_p$ whereas the "true groups" involve clustering $\mathbf{X}_1\beta_1, \ldots, \mathbf{X}_p\beta_p$. Therefore, the clusters estimated by CGL are not the optimal clusters for

ρ	Method	$\|\|\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\hat{\boldsymbol{\beta}}\|\|_2$	Correct Sparsity	Num. Non-Zeros	RI
0.0	Lasso	163.142(1.827)	0.869(0.009)	155.8(9.084)	0.907(0)
	Ridge	182.167(0.869)	0.05(0)	1000(0)	0.904(0.003)
	EN	162.428(1.875)	0.824(0.013)	204.5(13.951)	0.908(0)
	CEN	163.351(1.862)	0.807(0.026)	222.233(26.953)	0.908(0)
	CEN Known Groups	159.508(1.65)	0.815(0.01)	215.9(11.004)	1(0)
	Cluster Group Lasso	184.08(0.81)	0.05(0)	1000(0)	0.366(0)
	Group Lasso Known Groups	58.315(1.191)	0.05(0)	1000(0)	1(0)
0.1	Lasso	98.067(2.694)	0.914(0.005)	134.567(5.256)	0.953(0.003)
	Ridge	200.597(1.38)	0.05(0)	1000(0)	0.948(0.002)
	EN	97.396(2.472)	0.91(0.005)	138.9(5.478)	0.955(0.003)
	CEN	97.918(2.46)	0.868(0.029)	181.4(29.42)	0.954(0.003)
	CEN Known Groups	95.571(2.192)	0.877(0.012)	172.1(11.941)	1(0)
	Cluster Group Lasso	165.216(3.115)	0.131(0.03)	919(30.146)	0.397(0.005)
	Group Lasso Known Groups	41.672(0.844)	0.303(0.078)	746.667(78.012)	1(0)
0.2	Lasso	80.939(1.821)	0.938(0.004)	111.533(3.986)	0.979(0.002)
	Ridge	186.734(1.485)	0.05(0)	1000(0)	0.937(0.002)
	EN	80.194(1.766)	0.932(0.004)	117.867(4.159)	0.981(0.002)
	CEN	73.571(1.28)	0.716(0.025)	334.067(25.199)	0.984(0.001)
	CEN Known Groups	75.091(1.266)	0.804(0.016)	246.433(16.382)	1(0)
	Cluster Group Lasso	76.642(3.976)	0.08(0.03)	970.067(29.933)	0.84(0.023)
	Group Lasso Known Groups	37.685(0.805)	0.62(0.086)	430(86.423)	1(0)
0.5	Lasso	66.674(1.405)	0.955(0.003)	94.4(2.909)	0.982(0.001)
	Ridge	150.026(1.266)	0.05(0)	1000(0)	0.91(0.001)
	EN	65.067(1.304)	0.945(0.003)	104.267(2.843)	0.984(0.001)
	CEN	62.292(1.342)	0.814(0.038)	236.433(38.022)	0.988(0.001)
	CEN Known Groups	52.011(0.81)	0.773(0.027)	277.433(26.694)	1(0)
	Cluster Group Lasso	59.171(0.835)	0.08(0.03)	970(30)	0.906(0)
	Group Lasso Known Groups	30.612(0.711)	0.842(0.066)	208.333(65.744)	1(0)
0.8	Lasso	60.436(1.217)	0.955(0.003)	88.733(3.04)	0.964(0.002)
	Ridge	115.42(1.124)	0.05(0)	1000(0)	0.906(0)
	EN	53.421(0.818)	0.937(0.004)	112.467(3.632)	0.969(0.002)
	CEN	43.519(0.763)	0.75(0.047)	299.267(46.673)	0.991(0.001)
	CEN Known Groups	32.152(0.841)	0.78(0.046)	270.167(45.527)	1(0)
	Cluster Group Lasso	48.707(0.516)	0.23(0.067)	820(66.85)	0.906(0)
	Group Lasso Known Groups	21.592(0.603)	0.842(0.066)	208.333(65.744)	1(0)

Table 1: Simulation results. Means (and standard errors) over 30 iterations are reported. All models were fit on a training set using the set of tuning parameters that led to the smallest value of $||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}||_2$ on a validation set; in the third column, results on a separate test set are reported. The fourth column is the fraction of features correctly determined to be zero or non-zero, as defined in the beginning of Section 6.2. The sixth column is the Rand Index, which quantifies agreement between true and estimated clusters.

estimating the sparsity structure. For this reason, CGL performs far worse than group lasso with known groups, across all values of ρ.

CGL performs far worse than CEN when $\rho < 0.2$ (in which case k-means clustering of the features fails to identify the groups) and when $\rho = 0.8$ (in which case CEN more effectively

exploits the correlation structure within each group). For intermediate values of ρ, CEN and CGL perform comparably in terms of prediction error. CEN always outperforms CGL in terms of identification of the true clusters, as quantified by the Rand index.

6.3 Misspecification of the Number of Clusters

In Section 6.2, we performed CEN using $K = 3$, which is the true number of clusters of $X_j\beta_j$. However, in a typical application we will not know the true value for K. Table 2 displays the results obtained if CEN is performed using various values for K, under the simulation set-up described in Section 6.1 with $\rho = 0.5$. We find that though the best results are obtained for $K = 3$, competitive results are obtained for the other values of K considered. Given that CEN with $K = 1$ is essentially equivalent to the elastic net, and CEN with $K = p$ is simply the lasso, it is not surprising that using an intermediate though incorrect value of K can yield good results.

	$\|\|\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\hat{\boldsymbol{\beta}}\|\|_2$	Correct Sparsity	Num. Non-Zeros	RI
K=2	63.503(1.323)	0.931(0.008)	118.867(8.032)	0.948(0.002)
K=3	62.292(1.342)	0.814(0.038)	236.433(38.022)	0.988(0.001)
K=5	64.167(1.254)	0.853(0.024)	197.267(24.232)	0.988(0.001)
K=7	66.926(1.594)	0.899(0.02)	151.1(19.862)	0.985(0.002)

Table 2: The simulation set-up in Section 6.1 was performed with $\rho = 0.5$, and CEN models were fit using various values of K, using a training/validation/test set approach as described in Sections 6.1 and 6.2. Means (and standard errors) over 30 simulated data sets are reported. Column labels are as in Table 1.

7 Application to HIV Drug Resistance Data

We now consider the task of predicting the susceptibility of HIV-1 isolates to nucleoside reverse transcriptase inhibitor drugs (NRTIs) on the basis of the isolates' amino acid sequences (Rhee et al. 2006). A number of drugs are available for treating HIV-1 infection, and resistance to these drugs occurs due to mutations in the HIV-1 sequence. Being able to accurately predict susceptibility to a given drug based on the amino acid sequence of a given isolate would have important clinical implications, since then an individual can be given an optimal course of treatment based on the genotype of the HIV-1 isolates carried. A mutation in a particular

amino acid may lead to decreased drug susceptibility due to a change in that drug's binding site. In this case, nearby mutations may also lead to a similar change in the binding site.

For each of $n = 639$ HIV-1 isolates, the amino acid sequence of the first $p = 240$ positions in the reverse transcriptase gene is available from the Stanford HIV Drug Resistance Database, at `http://hivdb.stanford.edu/pages/published_analysis/genophenoPNAS2006/` (Shafer 2006). We coded each amino acid as a 0 in a given isolate if the wild-type amino acid is present, and a 1 otherwise. This resulted in a 639×240 binary data matrix. We do not expect there to be high correlations among the features in this data set, since the data are binary, and most isolates have just a few mutations. However, we do expect that adjacent mutations should have a similar association with the response, since mutations at two adjacent sites may have a similar effect on a particular drug's binding site.

In order to exploit our hypothesis that mutations at adjacent sites have similar associations with the response, we created a *pooled* data set by summing the mutations in a sliding window of five amino acids, resulting in a 639×236 data matrix. Adjacent features of this pooled data set are highly correlated with each other; furthermore, we expect adjacent features to have a similar association with the response. Features were scaled to have mean zero and unit variance. When used to predict susceptibility to didanosine, CEN and elastic net yielded similar test errors as evaluated using a training/validation/test set approach. However, for a range of values of K, the CEN models were much more interpretable, as is shown in Figure 5. CEN effectively assigns adjacent features to the same cluster, indicating that such features are highly correlated and have a shared association with the response. Similar results to Figure 5 were obtained using sliding windows of different widths.

As pointed out by a referee, on this data set there is a linear ordering among the features, and so the fused lasso (Tibshirani et al. 2005) is a natural alternative to CEN. In fact, applying the fused lasso to this data yields qualitatively similar results to applying CEN (though of course CEN does not set coefficient values to be exactly identical to each other). However, CEN is intended for settings in which the variables are unordered, or at least have no known ordering, so that fused lasso is not a possibility; we apply it to this data set with ordered features only to illustrate that the technique yields scientifically plausible results. We also note that in this application we could have analyzed the unpooled data; however, pooling

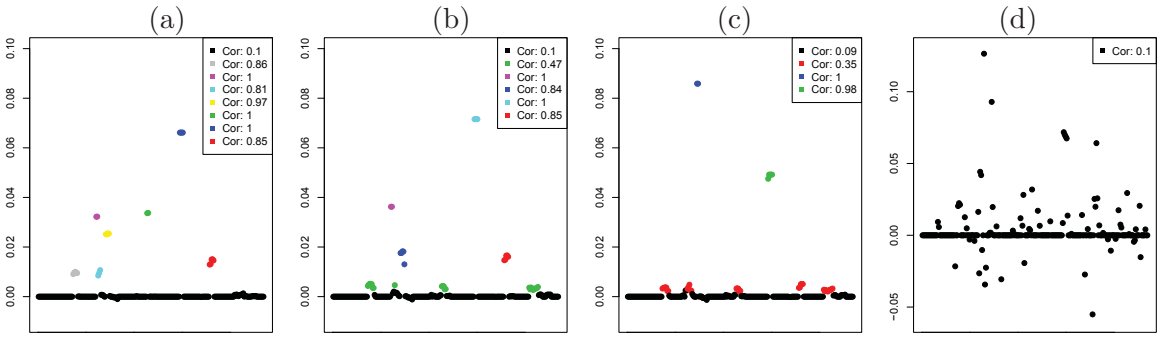

Figure 5: Coefficient estimates for prediction of susceptibility to the NRTI didanosine on the basis of pooled mutation status. The coefficient estimates are displayed for CEN with *(a):* $K = 8$, *(b):* $K = 6$, *(c):* $K = 4$, and *(d):* elastic net; recall that this is equivalent to CEN with $K = 1$ on a rescaled version of the data. For CEN, the coefficient estimates are colored by cluster label. The mean absolute pairwise correlation of the features within each cluster is displayed in the legend. CEN was performed with $\lambda = 20$ and $\delta = 45$ in each panel; elastic net was performed with equal tuning parameters on the ridge and lasso components, chosen to yield the average number of non-zero coefficients obtained in panels (a)-(c). Similar results were obtained in the models for predicting the other NRTIs, and using different sizes of sliding windows.

the features before performing the analysis leads to a nice interpretation of the non-zero coefficients in the resulting model as potential drug binding sites.

8 Discussion

In this paper, we have proposed the cluster elastic net, a technique for high-dimensional regression in the presence of unknown groups among the covariates. An efficient coordinate descent algorithm for solving the cluster elastic net optimization problem in the high-dimensional setting has been presented. We have shown that this procedure outperforms existing techniques under a range of simulated settings, and yields more interpretable results in an application to HIV drug sensitivity data.

We have discussed the use of the cluster elastic net in the least squares regression context. However, an extension to generalized linear models would be straightforward: it would simply entail applying the cluster elastic net penalty to the appropriate log likelihood function.

As was pointed out by a reviewer, in this paper we have not discussed the issue of inference for the coefficients in the cluster elastic net model. Indeed, inference in the high-dimensional setting is a challenging problem, and is currently a very active area of research. Existing

techniques may be applied or extended in order to address this problem (see e.g. Meinshausen & Buhlmann 2010, Berk et al. 2013, Lockhart et al. 2013).

Acknowledgments

An associate editor and two referees provided helpful comments that led to improvements in this paper. We thank Howard Bondell for providing `R` code for the PACS proposal. DW and FZ were supported by NIH Grant DP5OD009145. AS was supported by NSF Grant DMS-1161565.

Supplementary Materials

Title: Proof of Theorem 1.

References

Berk, R., Brown, L., Buja, A., Zhang, K. & Zhao, L. (2013), 'Valid post-selection inference', *Annals of Statistics* .

Bondell, H. & Reich, B. (2008), 'Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR', *Biometrics* **64**, 115–123.

Buhlmann, P., Rutimann, P., van de Geer, S. & Zhang, C. (2012), 'Correlated variables in regression: clustering and sparse estimation'.

Chiquet, J., Grandvalet, Y. & Charbonnier, C. (2012), 'Sparsity with sign-coherent groups of variables with the copperative-lasso', *Annals of Applied Statistics* .

Daye, Z. & Jeng, X. (2009), 'Shrinkage and model selection with correlated variables via weighted fusion', *Computational Statistics and Data Analysis* **53(4)**, 1284–1298.

Dettling, M. & Buhlmann, P. (2004), 'Finding predictive gene groups from microarray data', *Journal of Multivariate Analysis* **90**, 106–131.

Friedman, J., Hastie, T., Hoefling, H. & Tibshirani, R. (2007), 'Pathwise coordinate optimization', *Annals of Applied Statistics* **1**, 302–332.

Hastie, T., Tibshirani, R., Botstein, D. & Brown, P. (2001), 'Supervised harvesting of expression trees', *Genome Biology* **2(1)**, 1–12.

Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning; Data Mining, Inference and Prediction*, Springer Verlag, New York.

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

Huang, J., Ma, S., Li, H. & Zhang, C. (2011), 'The sparse Laplacian shrinkage estimator for high-dimensional regression', *The Annals of Statistics* **39(4)**, 2021–2046.

Jacob, L., Obozinski, G. & Vert, J. (2009), 'Group lasso with overlap and graph lasso', *Proceedings of the 26th International Conference on Machine Learning* .

Li, C. & Li, H. (2008), 'Network-constrained regularization and variable selection for analysis of genomic data', *Bioinformatics* **24**(9), 1175–1182.

Li, C. & Li, H. (2010), 'Variable selection and regression analysis for graph-structured covariates with an application to genomics', *Annals of Applied Statistics* **4(3)**, 1498–1516.

Lockhart, R., Taylor, J., Tibshirani, R. & Tibshirani, R. (2013), 'A significance test for the lasso'.

Meinshausen, M. & Buhlmann, P. (2010), 'Stability selection (with discussion)', *Journal of the Royal Statistical Society, Series B* **72**, 417–473.

Park, M. Y., Hastie, T. & Tibshirani, R. (2007), 'Averaged gene expressions for regression', *Biostatistics* pp. 212–227.

Rand, W. M. (1971), 'Objective criteria for the evaluation of clustering methods', *Journal of the American Statistical Association* **66**, 846–850.

Rhee, S., Taylor, J., Wadhera, G., Ben-Hur, A., Brutlag, D. & Shafer, R. (2006), 'Genotypic predictors of human immunodeficiency virus type 1 drug resistance', *Proceedings of the National Academy of Sciences* **103(46)**, 17355–17360.

Shafer, R. (2006), 'Rational and uses of a public HIV drug-resistance database', *Journal of Infectious Diseases* **194**, S51–8.

Sharma, D., Bondell, H. & Zhang, H. (2013), 'Consistent group identification and variable selection in regression with correlated predictors', *Journal of Computational and Graphical Statistics* .

She, Y. (2010), 'Sparse regression with exact clustering', *Electron. J. Statist.* **4**, 1055–1096.

Shen, X., Huang, H. & Pan, W. (2012), 'Simultaneous supervised clustering and feature selection over a graph', *Biometrika* .

Simon, N., Friedman, J., Hastie, T. & Tibshirani, R. (2010), 'A sparse-group lasso', *http://www-stat.stanford.edu/∼nsimon/SGLpaper.pdf* .

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *J. Royal. Statist. Soc. B.* **58**, 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. & Knight, K. (2005), 'Sparsity and smoothness via the fused lasso', *J. Royal. Statist. Soc. B.* **67**, 91–108.

Tutz, G. & Ulbricht, J. (2009), 'Penalized regression with correlation-based penalty', *Statistics and Computing* **19**, 239–253.

Yuan, M. & Lin, Y. (2007), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society, Series B* **68**, 49–67.

Zou, H. & Hastie, T. (2005), 'Regularization and variable selection via the elastic net', *J. Royal. Stat. Soc. B.* **67**, 301–320.