Journal of Statistical Research *https://doi.org/10.3329/jsr.v56i1.63945*
2022, Vol. 56, No. 1, pp. 37-53 *ISSN 0256 - 422 X*

SPARSE REGRESSION WITH CLUSTERED PREDICTORS

AIDEN KENNY

Franklin Marshall College, P.O. Box 3003, Lancaster, PA 17604-3003, USA

Email: akeny@fandm.edu

DANIELLE SOLOMON

St. Johns University, 8000 Utopia Parkway, Queens, NY 11439, USA

Email: danielle.solomon17@my.stjohns.edu

MOHAMMAD S. A. PATWARY

Butler University, 4600 Sunset Ave, Indianapolis, IN 46208, USA

Email: mpatwary@butler.edu

KUMER P. DAS⋆

University of Louisiana at Lafayette, 104 E University Ave, Lafayette, LA 70504, USA

Email: kumer.das@louisiana.edu

SUMMARY

Gene expression data can be challenging to analyze due to its high-dimensional nature. Regularization techniques are useful in reducing the number of predictors and highlighting the significant genes, in this case, genes that may indicate the presence of cancer. This study aims to see if grouping the genes before applying the regularization techniques is beneficial in reducing the prediction error of classification. We investigate the potential effectiveness of using clustering algorithms to generate a grouping structure for high-dimensional data sets. Using various regularization techniques, we seek to determine if the generated groups are truly relevant to the response and if the accuracy and interpretability of the models can be improved. We apply the clustered group structure to two real-world data sets. We also employ simulation studies to assess the performance of different regularization methods for both clustering and no-clustering methods.

Keywords and phrases: High dimensional data; Logistic regression; Sparse regression; Regularization; Principal Component Analysis; Clustering.

AMS Classification: 62H25, 62J99, 62N86

⋆ Corresponding author

1 Introduction

The idea of using data and information to train models that are both accurate and interpretable has been around for decades.Very recently, Lee et al. (2018) explored different analysis techniques for microarray data to create a more effective predictor of age from DNA methylation level. One desires to build a model based on the *predictors* that is both accurate and interpretable; we want our models to predict the outcome correctly, and we want to know which predictors are responsible. However, in the age of big data, it is becoming increasingly common that a data set is *high-dimensional*, meaning the number of predictors p vastly exceeds the number of observations $n.$ In this setting, many longstanding statistical modeling techniques, such as linear and logistic regression, no longer suffice. Regularization is a popular technique that imposes a penalty on the original model; in some cases, the models are *sparse*, meaning they are very interpretable.

Sometimes, the predictors of a model belong to some kind of pre-defined group, and the response is based on these groups, as opposed to the individual predictors. More advanced regularization methods have been developed to accommodate group structure, and assuming that the groups are well-represented can greatly improve the accuracy and interpretability of the models. Unfortunately, while the response could truly be dependent on the group structure, the actual grouping structure is unknown beforehand. In this situation, one would desire to properly identify the grouping structure and build a model based on the result.

This paper aims to investigate the effect that clustering can have on regularized models. We seek to answer two questions:

1. Can clustering algorithms be used to properly identify a grouping structure in a data set?
2. Can grouping the predictors using the clustering information improve the accuracy and interpretability of the model?

The rest of the article is organized into five sections. In Section 2 we provide a brief overview of the various regularization techniques and clustering algorithms we used in our study. Sections 3 and 4 investigate the effect of clustering predictors on two real-world genomic data sets, Section 5 provides two scenarios of simulation study, and we conclude with a discussion in Section 6.

2 Methodology

2.1 Logistic regression

In many situations, the response variable of a data set is categorical in nature, and we wish to assign an observation to one of the response variables given its inputs, a process known as classification. We seek to model the *probability* that an observation falls into a given class. It is often the case where the response belongs to one of two classes coded as $\mathcal{G} = \{0, 1\}$. In this binary setting, one popular approach to modeling the probabilities is *logistic regression*.

Suppose we have n observations and p predictors stored in a data matrix $\mathbf{X} = \{x_{i,j}\}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$, along with a response vector $\mathbf{y} = (y_1, \ldots, y_n)'$, where $y_i \in \{0, 1\}$.

If $p(\boldsymbol{x}_i) = \mathbb{P}(Y = 1 \mid X = \boldsymbol{x}_i)$, where $\boldsymbol{x}_i = (x_{i,1}, \dots, x_{i,p})'$ is the ith observation in $\mathbf{X}$, then the probability is modeled (as the log-odds) by

$$\log\left(\frac{p(\boldsymbol{x}_i)}{1 - p(\boldsymbol{x}_i)}\right) = \beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta}. \tag{2.1}$$

From this, the estimated response $\hat{y}_i$ is 1 if $p(\boldsymbol{x}_i) \geq 0.5$ and 0 otherwise. The coefficients β_0 and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)$ are estimated from the data by minimizing the negative of log-likelihood function

$$L(\beta_0, \boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\Big[\log\Big(1 + e^{\beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta}}\Big) - y_i\big(\beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta}\big)\Big]. \tag{2.2}$$

2.2 Basic regularization

In general, a regularized linear model seeks to minimize a penalized version of (2.2) of the form

$$Q(\beta_0, \boldsymbol{\beta}) = L(\beta_0, \boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta}),$$

where $P(\boldsymbol{\beta})$ is some type of penalty imposed on the coefficient vector $\boldsymbol{\beta}$. The tuning parameter $\lambda \geq 0$ effectively controls the severity of the penalty; as the value of λ increases, more shrinkage is imposed on the coefficients.

Various regularization methods have been introduced throughout the years using different penalty functions, with each method shrinking the coefficients in a different way. *Ridge regression* (Hoerl and Kennard, 1970) imposes a squared ℓ_2 norm on $\boldsymbol{\beta}$, and seeks to minimize

$$Q(\beta_0, \boldsymbol{\beta}) = L(\beta_0, \boldsymbol{\beta}) + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2. \tag{2.3}$$

The ℓ_2 norm causes continuous shrinkage of the estimated coefficients. A major drawback to ridge regression is that it produces dense models, i.e. models where $\beta_j \neq 0$ for all j, an undesirable characteristic for an interpretable model.

An alternative similar to ridge regression is the *lasso* (Tibshirani, 1996), which minimizes

$$Q(\beta_0, \boldsymbol{\beta}) = L(\beta_0, \boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1. \tag{2.4}$$

Here, an ℓ_1 norm is imposed on $\boldsymbol{\beta}$, as opposed to a squared ℓ_2 norm. Unlike ridge regression, the lasso is able to perform variable selection, forcibly setting many estimated coefficients to zero, producing sparse models. The resulting sparsity of the model often makes the lasso more preferable than ridge regression in the high-dimensional setting. Unfortunately, the lasso has several caveats as well; in the high-dimensional setting the lasso will select at most n predictors, and if several predictors are highly-correlated, the lasso will select only one and force the others to zero.

A generalization to ridge regression and the lasso, which attempts to combine the benefits while negating the drawbacks, is the *elastic net*[1] Zou and Hastie (2005), which minimizes

$$Q(\beta_0, \boldsymbol{\beta}) = L(\beta_0, \boldsymbol{\beta}) + \lambda\Big[\alpha\|\boldsymbol{\beta}\|_1 + \frac{1-\alpha}{2}\|\boldsymbol{\beta}\|_2^2\Big]. \tag{2.5}$$

[1] (Zou and Hastie, 2005) call this penalty the *naïve* elastic net penalty, and suggest that scaling the estimated coefficients up by a factor of $1 + \lambda(1 - \alpha)$ improves prediction accuracy. However, in their paper describing the implementation of the elastic net, Friedman et al. (2010) abandon this distinction.

This penalty is a linear combination of (2.3) and (2.4), and the mixing parameter $\alpha \in [0, 1]$ is used to determine how much of each type of penalty is imposed on the model; $\alpha = 0$ corresponds to ridge regression, while $\alpha = 1$ gives the lasso.

2.3 The group setting

Much work has been done to develop penalties that exploit pre-determined group structure. Suppose that the predictors of $\mathbf{X}$ are split into K non-overlapping groups, with S_k denoting the size of the kth group. For $k = 1, \ldots, K$, let $\mathbf{X}_k \in \mathbb{R}^{n \times S_k}$ denote the data matrix with the predictors in group k, and let $\boldsymbol{\beta}_k = (\beta_{k,1}, \ldots, \beta_{k,S_k})$ be the sub-vector of $\boldsymbol{\beta}$ corresponding to the kth group.

The *group lasso* ("gLasso") (Yuan and Lin, 2006) imposes an ℓ_2 norm on each of the coefficient sub-vectors; it minimizes

$$Q(\beta_0, \boldsymbol{\beta}) = L(\beta_0, \beta) + \lambda \sum_{k=1}^{K} \sqrt{S_k} \|\boldsymbol{\beta}_k\|_2. \tag{2.6}$$

The group lasso was later extended to logistic regression by Meier et al. (2008). The ℓ_2 penalties on each of the coefficient sub-vectors creates sparsity among the different groups while performing ridge shrinkage within each group. As a result, the group lasso unfortunately only induces sparsity at the group level, and if a group is determined to be significant, *all* of the group's predictors will be nonzero.

Both Yuan and Lin (2006) and Meier et al. (2008) assume that the data is orthonormal within each group, i.e. $\mathbf{X}_k^T \mathbf{X}_k = \mathbf{I}$ for all k. This is almost never the case in practice, so one would want to orthonormalize each $\mathbf{X}_k$ before minimizing (2.6). However, as Simon and Tibshirani (2012) show, this actually changes the penalty to

$$Q(\beta_0, \boldsymbol{\beta}) = L(\beta_0, \beta) + \lambda \sum_{k=1}^{K} \sqrt{S_k} \|\mathbf{X}_k \boldsymbol{\beta}_k\|_2. \tag{2.7}$$

This alternative penalty is theoretically and computationally superior (Breheny and Huang, 2015) to (2.6), so for the rest of this paper we refer to (2.7) when speaking about the group lasso.

There are several methods used in practice to induce sparsity both within and among groups, a feature known as *bi-variate selection*. One method is to combine the lasso and the group lasso penalties as a linear combination, similar to the elastic net; the resulting penalty is known as the the *sparse group lasso* Simon et al. (2013), and minimizes

$$Q(\beta_0, \boldsymbol{\beta}) = L(\beta_0, \boldsymbol{\beta}) + \lambda \Big[\alpha \|\boldsymbol{\beta}\|_1 + (1 - \alpha) \sum_{k=1}^{K} \sqrt{S_k} \|\boldsymbol{\beta}_k\|_2 \Big]. \tag{2.8}$$

With this penalty, sparsity is induced at the group level, and elastic net-type shrinkage is imposed within each group. Unfortunately, unlike the group lasso, there is no way to orthonormalize each group without corrupting the within-group sparsity effect, making any implementation of the sparse group lasso difficult compared to other penalties.

An alternative method for performing bi-variate selection is the *composite minimax concave penalty*[2] ("cMCP") (Breheny and Huang, 2009), which minimizes

$$Q(\beta_0, \boldsymbol{\beta}) = L(\beta_0, \boldsymbol{\beta}) + \sum_{k=1}^{K} f_{\lambda,\Gamma_k}\left(\sum_{s=1}^{S_k} f_{\lambda,\gamma}(|\beta_{k,s}|)\right). \tag{2.9}$$

Here $f_{\lambda,\gamma}(\cdot)$ is the minimax concave penalty (Zhang, 2007), given by

$$f_{\lambda,\gamma}(\phi) = \begin{cases} \lambda\phi - \frac{\phi^2}{2\gamma}, & \text{if } \phi \leq \gamma\lambda, \\ \frac{1}{2}\gamma\lambda^2, & \text{if } \phi > \gamma\lambda. \end{cases} \tag{2.10}$$

The intuition behind (2.10) is to counter the aggressive shrinkage that the lasso imposes on large coefficients. The parameter $\gamma > 0$ controls the "range" of this counter, and the penalty becomes the lasso as $\gamma \to \infty$. We can see that (2.9) is effectively applying the penalty twice, once to induce sparsity within each group and then again to induce sparsity among the groups. The outer parameter is set to $\Gamma_k = \frac{1}{2}S_k\gamma\lambda$, while the inner penalty γ is the same throughout.

2.4 Regularization based on principal components

Let $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ be the singular value decomposition of the data matrix, and let $m = \text{rank}(\mathbf{X})'$. The principal axes, or right singular vectors, are given by the columns of $\mathbf{V} \in \mathbb{R}^{p\times m}$, and $\boldsymbol{d} = (d_1, \ldots, d_m)$ are the singular values such that $d_1 \geq \ldots \geq d_m > 0$. $\mathbf{D} \in \mathbb{R}^{m\times m}$ is a diagonal matrix whose diagonal entries are the elements of $\boldsymbol{d}$.

Principal components lasso ("pcLasso") (Tay et al., 2018) minimizes

$$Q(\beta_0, \boldsymbol{\beta}) = L(\beta_0, \boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1 + \frac{\theta}{2}\boldsymbol{\beta}^T\left(\mathbf{V}\mathbf{D}_{d_1^2-d_j^2}\mathbf{V}^T\right)\boldsymbol{\beta}, \tag{2.11}$$

where λ and θ are two separate tuning parameters. The diagonal matrix $\mathbf{D}_{d_1^2-d_j^2} \in \mathbb{R}^{m\times m}$ has diagonal inputs that are functions of the singular values of $\mathbf{X}$, and is given by

$$\mathbf{D}_{d_1^2-d_j^2} = \text{diag}(d_1^2 - d_1^2, d_1^2 - d_2^2, \ldots, d_1^2 - d_m^2). \tag{2.12}$$

This "pcLasso penalty" has the result of imposing less shrinkage in the direction of the leading principal axis and more severe shrinkage in the directions of subsequent principal axes. In other words, $\boldsymbol{\beta}$ is biased in the direction of the leading principal axis. The presence of the ℓ_1 norm allows pcLasso to simultaneously perform feature selection.

pcLasso can also be modified to exploit group structure. Let $\mathbf{X}_k = \mathbf{U}_k\mathbf{D}_k\mathbf{V}_k^T$ be the singular value decomposition for the kth group matrix, and let $m_k = \text{rank}(\mathbf{X}_k)$. Then the columns of $\mathbf{V}_k$ and $\boldsymbol{d}_k = (d_{k,1}, \ldots, d_{k,m_k})$ are the principal axes and singular values of $\mathbf{X}_k$, respectively. In this

[2] Breheny and Huang (2009) originally denote cMCP as the *group* MCP. To avoid confusion, Huang et al. (2012) recommend denoting (2.9) as the *composite* MCP.

setting, pcLasso seeks to minimize

$$Q(\beta_0,\boldsymbol{\beta}) = L(\beta_0,\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1 + \frac{\theta}{2}\sum_{k=1}^{K}\sqrt{S_k}\boldsymbol{\beta}_k^T\Big(\mathbf{V}_k\mathbf{D}_{d_{k,1}^2-d_{k,j}^2}\mathbf{V}_k^T\Big)\boldsymbol{\beta}_k. \tag{2.13}$$

Similar to (2.12), the matrix $\mathbf{D}_{d_{k,1}^2-d_{k,j}^2} \in \mathbb{R}^{m_k\times m_k}$ is given by

$$\mathbf{D}_{d_{k,1}^2-d_{k,j}^2} = \mathrm{diag}(d_{k,1}^2 - d_{k,1}^2, d_{k,1}^2 - d_{k,2}^2, \ldots, d_{k,1}^2 - d_{k,m_k}^2). \tag{2.14}$$

We now see that pcLasso biases each coefficient sub-vector $\boldsymbol{\beta}_k$ in the direction of that group's leading principal axis, all while producing sparse models. Unlike the group lasso and cMCP, pcLasso does not require each group matrix $\mathbf{X}_k$ to be orthonormal.

2.5 Clustering methods

In general, a clustering algorithm seeks to partition the predictors of a data set into different subgroups based on some dissimilarity measure; ideally, the dissimilarity will be low for predictors within the same cluster and high for predictors in separate clusters. Various clustering algorithms and dissimilarity measures exist that seek to achieve this goal, so for this report we only focus on two simple algorithms.

K-means clustering (MacQueen et al., 1967) clusters the predictors into K non-overlapping groups based on their Euclidean distance in the observation space. Let C_k be the set of all predictors that belong to group k, for $k = 1,\ldots,K$, and let S_k denote the number of predictors in group k. K-means clustering seeks to minimize the total within-cluster variation, given by

$$W_K = \sum_{k=1}^{K}\sum_{j\in C_k}\|\mathbf{x}_j - \bar{\mathbf{x}}_k\|_2^2, \tag{2.15}$$

where $\mathbf{x}_j$ is the jth predictor and $\bar{\mathbf{x}}_k = \frac{1}{S_k}\sum_{j\in C_k}\mathbf{x}_j$ is the kth group centroid.

One drawback of K-means clustering is that the number of clusters K must be supplied before W_K can be minimized. Given that we have no information about the group structure beforehand, we desire some type of measurement to determine the optimal number of clusters. The GAP statistic (Tibshirani et al., 2001), defined as

$$\mathrm{Gap}(m) = \mathbb{E}\left[\log(W_m)\right] - \log(W_m), \tag{2.16}$$

attempts to choose an optimal K from the rate that W_K decreases. Given a maximum amount of clusters M, the gap statistic is estimated using B Monte Carlo random samples; the optimal number of clusters K is chosen when $\mathrm{Gap}(K) \geq \mathrm{Gap}(K+1) - \delta(K+1)$, where

$$\delta(m) = \mathrm{sd}_m\sqrt{1+\frac{1}{B}}$$

and sd_m is the standard deviation of the B estimated W_m's.

Another appealing clustering algorithm is *hierarchical clustering*, which groups the predictors into nested clusters. While Euclidean distance could be used as the choice of dissimilarity, we chose to use correlation instead to investigate how this choice effects the resulting group structure. Unlike K-means clustering, there is no criteria that could be used to choose the optimal number of clusters. Fortunately, given that the clusters are nested, they can be represented in a dendrogram where the number of clusters can be chosen by the user.

3 Colon Data Set

The colon data set, originally introduced by Alon et al. (1999), contains the gene expressions of 2,000 genes for 62 different tissue samples, i.e. $n = 62$ and $p = 2,000$. Of the 62 tissue samples, 40 of the samples tested positive for colon cancer, while 22 tested negative.

3.1 Clustering information

Figure 1 shows the clustering information for the colon data set. The top-left panel measures the gap statistics for $m = 1, \ldots, 20$, and chooses $K = 9$ as the optimal number of clusters using K-means clustering. The top-right panel plots the predictors against the first two columns of $\mathbf{Z} = \mathbf{VD}$ (where $\mathbf{VDU}^T = \mathbf{X}^T$ is the SVD of the transposed data matrix, so the columns of $\mathbf{Z}$ are the principal components of $\mathbf{X}^T$), along with the labeled groups that each predictor belongs to.

The bottom panel displays the corresponding dendrogram using correlation as the dissimilarity measure for hierarchical clustering. We decided that $K = 7$ was a reasonable cut-off for this dendrogram.

3.2 Results

For all of the following methods, we randomly split the data set into a training set and a test set, both with 31 observations each. Each model was fit on the training set, and its performance was measured on the test set. Models that have additional parameters besides λ were fit using a grid of values of said parameter (e.g. α for the elastic net), and the model with the lowest deviance was chosen.

We first fit the following regularized models on the colon data set *without* any grouping structure:

1. The lasso: was fit using `glmnet` version `2.0.16`.
2. The elastic net: was fit for $\alpha = 0.95, 0.8, 0.6, 0.4, 0.2, 0.05$, using `glmnet` version `2.0.16`.
3. pcLasso: was fit for `rat`[3] $= 0.95, 0.9, 0.75, 0.5, 0.25, 0.1$, using `pcLasso` version `1.1`.

Next, we fit the following models on the colon data set using the grouping structure obtained from K-means clustering:

[3] As opposed to testing over a grid of values for θ, Tay et al. (2018) suggest specifying a value of `rat` instead. More details can be found in their paper.

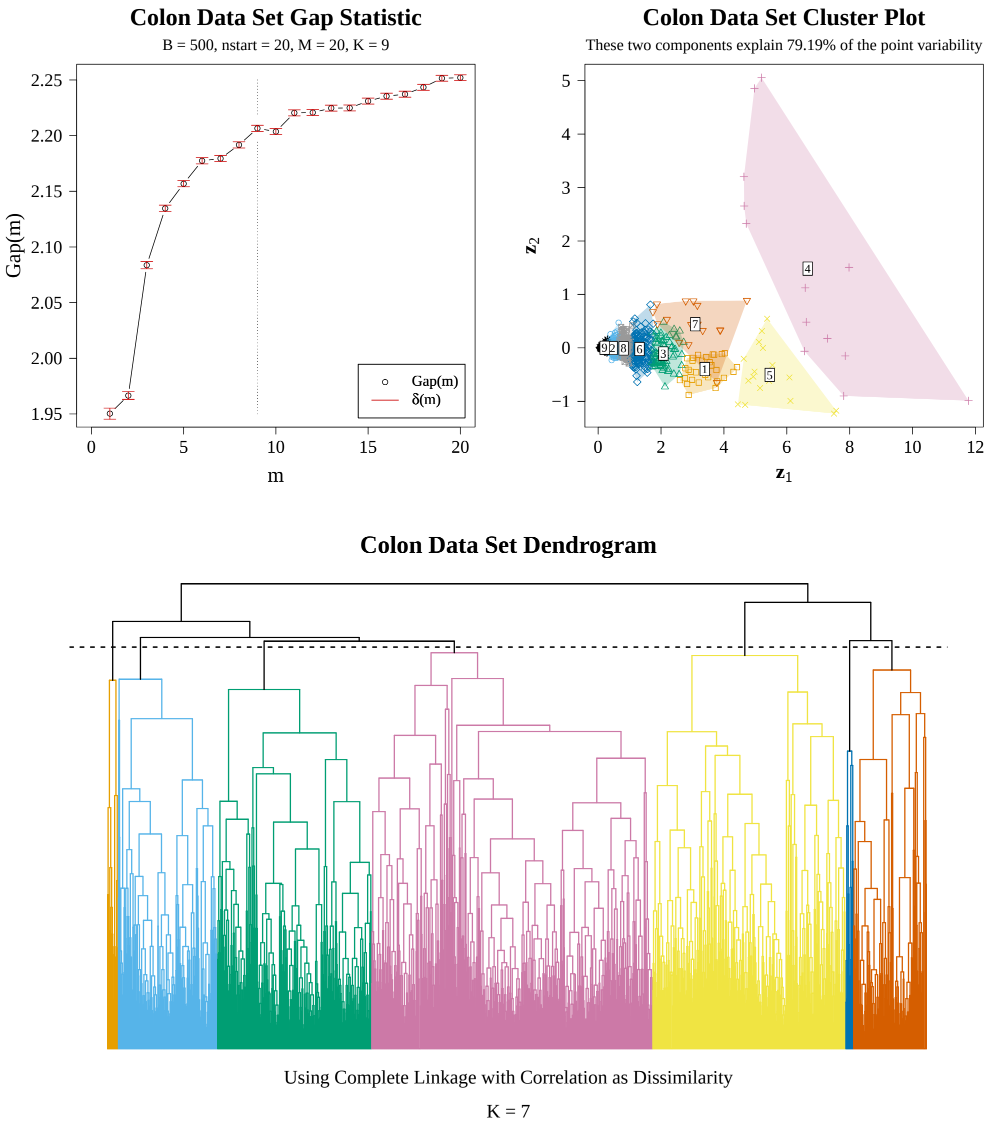

Figure 1: Clustering information for the colon data set.

1. gLasso: was fit using `grpreg` version `3.2.1`.

2. sgLasso: was fit for $\alpha = 0.95, 0.8, 0.6, 0.4, 0.2, 0.05$, using `SGL` version `1.2`.

3. cMCP: was fit for $\gamma = 30$, using `grpreg` version `3.2.1`.

4. pcLasso: was fit for `rat` $= 0.95, 0.9, 0.75, 0.5, 0.25, 0.1$, using `pcLasso` version `1.1`.

The `clusGap` function from `cluster` version `2.1.0` was used to calculate the gap statistic, and the groups were clustered using the `kmeans` function. Finally, the process above was repeated using the grouping structure from hierarchical clustering, which was obtained using the `hclust` function.

The results from the various models are presented in Table 1. Included in the table are the values of the optimized parameters, the cross-validation deviance, the number of missclassifications on the test set, the number of nonzero coefficients in the final model (including the intercept β_0), the number of significant groups in the final model (if a single group contained a nonzero coefficient, then it is considered significant), and the area under the curve (AUC) measurements. The corresponding ROC curves for each model have been printed in Figure 2.

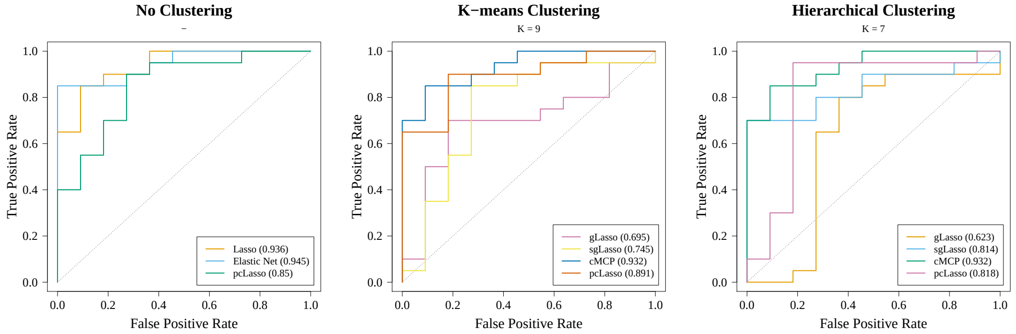

Figure 2: The ROC curves for the colon data set.

We can see that pcLasso using hierarchical clustering performs the best in terms of missclassifications, having only incorrectly predicting 3 of the 31 test observations. When looking at the missclassifications for all of the models, we can see that the lasso, the elastic net, and pcLasso without clustering all perform about the same.

With K-means clustering, gLasso missclassifies 12 of the 31 test observations, which is worse than a null model (the test set had 11 observations that tested negative for cancer and 20 that tested positive). sgLasso performs slightly worse, and cMCP and pcLasso perform about the same as the non-clustered models. Looking at the number of significant groups, we can see that pcLasso does not perform bi-variate selection at all, since all nine groups are represented in the final model. This, along with the poor performance of gLasso and sgLasso, indicate that the grouping structure obtained using K-means clustering is not sufficient.

A similar situation occurs with hierarchical clustering. This time we see that sgLasso missclassifies the most test observations, and gLasso also performs worse than the non-grouped models. Interestingly, the trained cMCP model using hierarchical clustering is identical in size to the model using K-means clustering. Finally, pcLasso here both missclassifies the least amount of test observations and has the least number of significant coefficients. There are six significant coefficients (keep in mind the table includes the intercept term) as well as six non-zero coefficients, so again pcLasso does not induce shrinkage at the group level. It is worth noting that pcLasso using hierarchical clustering also has less significant coefficients than all of the non-clustered models,

Table 1: The performance of various models on the colon data set.

	No Clustering			K-means Clustering				Hierarchical Clustering			
	–			$K = 9$				$K = 7$			
	Lasso	Elastic Net	pcLasso	gLasso	sgLasso	cMCP	pcLasso	gLasso	sgLasso	cMCP	pcLasso
Parameters	$\lambda = 0.0642$	$\lambda = 0.0853$	$\lambda = 8.75$	$\lambda = 0.00562$	$\lambda = 0.0215$	$\lambda = 0.0838$	$\lambda = 35.32$	$\lambda = 0.0621$	$\lambda = 0.0194$	$\lambda = 0.0813$	$\lambda = 163.94$
	–	$\alpha = 0.95$	$\texttt{rat} = 0.95$	–	$\alpha = 0.4$	$\gamma = 30$	$\texttt{rat} = 0.25$	–	$\alpha = 0.05$	$\gamma = 30$	$\texttt{rat} = 0.95$
Deviance	0.938	0.945	0.561	0.490	0.853	1.01	0.525	0.987	0.863	0.995	0.548
Misclass.	6/31	6/31	5/31	12/31	7/31	6/31	5/31	8/31	11/31	6/31	3/31
Sig. Coef.	16	19	30	49	29	11	30	20	461	11	7
Sig. Groups	–	–	–	3	2	4	9	1	1	4	6
AUC	0.936	0.945	0.850	0.695	0.745	0.932	0.891	0.623	0.814	0.932	0.818

so it is both more accurate and more interpretable.

4 Leukemia Data Set

The leukemia data set, from Golub et al. (1999), contains the gene expressions of 7,128 genes for 72 different patients ($n = 72$ and $p = 7,128$). The response is the type of leukemia each patient has; 47 were diagnosed with acute lymphoblastic leukemia (ALL) while 25 were diagnosed with acute myeloid leukemia (AML).

4.1 Clustering information

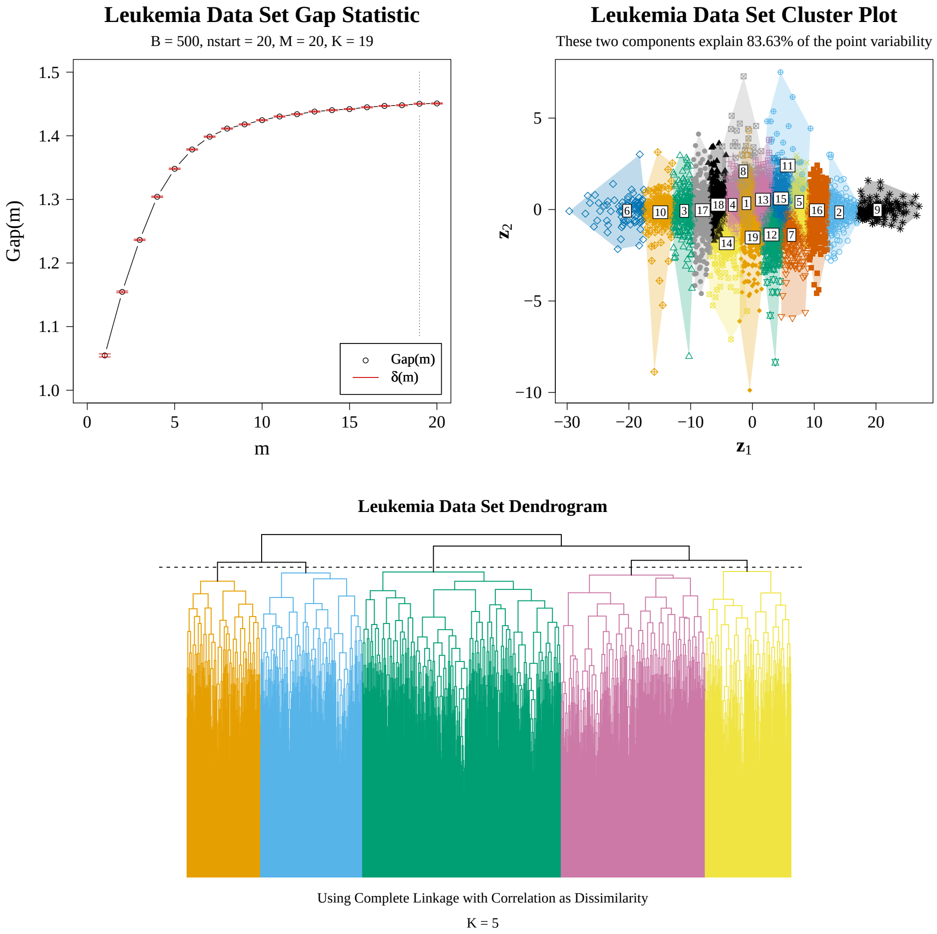

Figure 3: Clustering information for the leukemia data set.

The clustering information for the leukemia data set has been printed in Figure 3. The gap statistics for $m = 1, \ldots, 20$ have been printed in the top left panel, and $K = 19$ is chosen as the optimal number of clusters. The top right panels plots the predictors against the first two principal components of the observation space, with each group labeled. Finally, the bottom panel shows the leukemia dendrogram; we chose $K = 5$ at the optimal number of clusters.

4.2 Results

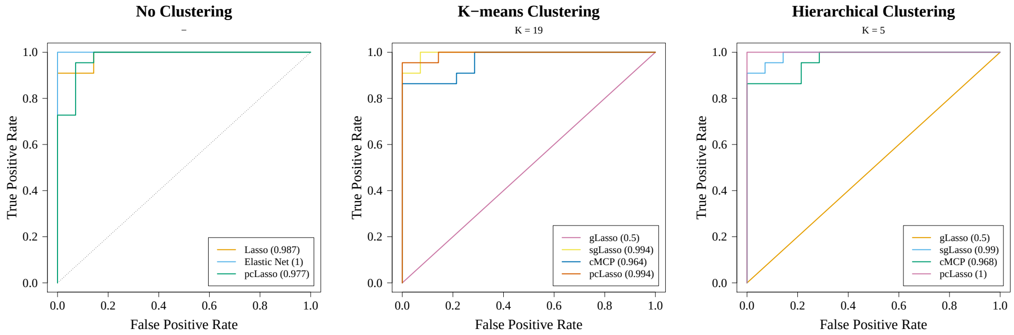

Figure 4: The ROC curves for the leukemia data set.

The same general set up for the colon data set was applied to the leukemia data set; the results have been printed in Table 2 and the ROC curves have been printed in Figure 4. As with the colon data set, it seems that the clustering algorithms are unable to identify a sufficient grouping structure.

For the non-clustered models, both the elastic net and pcLasso perform the same in terms of missclassifications. However, the elastic net has a markedly lower deviance, less significant coefficients, and a higher AUC, making it a decisively better model in this situation.

Both gLasso and sgLasso perform extremely poorly for both K-means and hierarchical clustering; in fact, gLasso actually fits a null model in both cases. cMCP again fits models with similar sizes in both cases, and while the missclassification rate is much better than gLasso and sgLasso, it still performs worse than the non-clustered models. As with the colon data set, clustered pcLasso is the overall winner, having only missclassified two of the 36 test observations in both cases. In addition, we see that pcLasso does not induce shrinkage at the group level. And while both models perform the same in prediction accuracy, pcLasso using hierarchical clustering has a slightly lower deviance, significantly less significant predictors, and a slightly higher AUC, so it can be considered the best model for all of the clustered predictors.

Unlike the colon data set, however, there is some type of trade-off that one would have to consider when choosing an overall best model. While pcLasso with hierarchical clustering has the lowest number of missclassifcations, it has 45 significant predictors. On the other hand, the lasso and the elastic net have 13 and 27, respectively, so even though their missclassification rates are slightly higher, they are more interpretable models. It is entirely reasonable for one to choose a model with one additional incorrect prediction if it is easier to interpret and explain.

Table 2: The performance of various models on the leukemia data set.

	No Clustering			K-means Clustering				Hierarchical Clustering			
	–			$K = 19$				$K = 5$			
	Lasso	Elastic Net	pcLasso	gLasso	sgLasso	cMCP	pcLasso	gLasso	sgLasso	cMCP	pcLasso
Parameters	$\lambda = 0.00407$	$\lambda = 0.00509$	$\lambda = 0.00548$	$\lambda = 0.0779$	$\lambda = 0.0457$	$\lambda = 0.111$	$\lambda = 0.00548$	$\lambda = 0.0779$	$\lambda = 0.0236$	$\lambda = 0.111$	$\lambda = 0.00548$
	–	$\alpha = 0.8$	$\texttt{rat} = 0.95$	–	$\alpha = 0.95$	$\gamma = 30$	$\texttt{rat} = 0.9$	–	$\alpha = 0.4$	$\gamma = 30$	$\texttt{rat} = 0.95$
Deviance	0.241	0.0834	0.465	1.240	0.730	0.671	0.442	1.240	0.731	0.671	0.439
Misclass.	5/36	3/36	3/36	14/36	13/36	6/36	2/36	14/36	14/36	6/36	2/36
Sig. Coef.	14	28	41	1	6	6	76	1	772	7	46
Sig. Groups	–	–	–	0	1	2	19	0	1	2	5
AUC	0.987	1.000	0.977	0.500	0.994	0.964	0.994	0.500	0.990	0.968	1.000

5 Simulation

To assess the performance of different regularization methods for both no-clustering and clustering methods we have simulated data for n = 100 subjects and $n.sim$ = 30 simulation runs were performed for the convenience of computation time. Keeping the fact in mind that we have predictors in real data analysis, we simulated P = 50 predictors from multivariate normal distributions with two different scenario (presented in the subsequent subsections). We have simulated the random coefficients for respective predictors from the continuous grid -3 and +3 of the uniform distribution. As performance metrics, we have observed deviance, misclassification rate, number of significant coefficients, and AUC across different clustering methods and respective regularization methods under each clustering method. In hierarchical clustering, we cut the tree at the height of 1.5 which is aligned with colon dataset and leukemia dataset. Since, we have performed 30 simulation runs, each of the performance metrics was averaged out but we have rounded misclassifcation rate to its nearest whole number to avoid confusion in understanding.

5.1 Scenario 1

In this case, we have simulated the predictors from a $N_P(\mathbf{0}, \mathbf{I})$, meaning that the data is not clustered. Using the logistic probability expression, $p = \big(1 + e^{-\mathbf{X}\boldsymbol{\beta}}\big)^{-1}$, we have simulated the success and failure probabilities. Using these probabilities, we have simulated the 100 random Bernoulli trials of successes and failures. Simulation results are presented in Table 3.

Table 3: The performance of various models on the non-clustered simulated dataset

	No Clustering			K-means Clustering				Hierarchical Clustering			
	–			$K = 1$				$K = 2$			
	Lasso	Elastic Net	pcLasso	gLasso	sgLasso	cMCP	pcLasso	gLasso	sgLasso	cMCP	pcLasso
Deviance	1.8924	1.3266	0.7983	1.9004	1.3259	0.8017	0.6922	1.5691	1.1257	0.7655	0.5267
Misclass.	5/100	5/100	4/100	10/100	8/100	7/100	4/100	9/100	11/100	6/100	4/100
Sig. Coef.	20	26	33	35	31	17	26	32	39	29	11
AUC	0.9567	0.9258	0.8239	0.6289	0.8658	0.9763	0.8029	0.7923	0.8547	0.9871	0.8892

5.2 Scenario 2

In this case, we have simulated the first 50 observations of predictors from a $N_P\big(-3\mathbf{1}_p, \mathbf{I}\big)$ and another 50 observations of predictors from $N_P\big(3\mathbf{1}_p, \mathbf{I}\big)$, meaning that the data is now clustered, which distinguish this scenario from scenario 1. Like as before, Using the logistic probability expression, $p = \big(1 + e^{-\mathbf{X}\boldsymbol{\beta}}\big)^{-1}$, we have simulated the success and failure probabilities. Using these probabilities, we have simulated the 100 random Bernoulli trials of successes and failures. Simulation results are presented in Table 4.

Table 4: The performance of various models on the clustered simulated dataset

	No Clustering			K-means Clustering				Hierarchical Clustering			
	–			$K = 2$				$K = 3$			
	Lasso	Elastic Net	pcLasso	gLasso	sgLasso	cMCP	pcLasso	gLasso	sgLasso	cMCP	pcLasso
Deviance	1.2357	0.9231	0.5677	1.0592	0.7981	0.6252	0.4669	1.0029	0.7887	0.5923	0.4169
Misclass.	7/100	7/100	5/100	12/100	10/100	8/100	7/100	11/100	10/100	7/100	6/100
Sig. Coef.	23	28	32	38	32	20	28	36	44	32	12
AUC	0.9723	0.9515	0.8970	0.7257	0.9012	0.9899	0.7789	0.8023	0.8455	0.9918	0.8720

6 Discussion

With both data sets, we saw that the group lasso and the sparse group lasso performed very poorly relative to the other methods, especially for the leukemia data set. In addition, pcLasso did not perform bi-variate selection. Both of these facts are indications that both K-means clustering and hierarchical clustering were *ineffective* in properly identifying a grouping structure that was relevant to the response. However, we also observed that pcLasso with clustered predictors had a slighlty lower number of missclassifications than the non-clustered models, showing that even though the groups were not relevant to the response, the models can potentially become more accurate and interpretable.

From the results presented in Table 3 and Table 4 for non-clustered and clustered simulated data, respectively, it has been observed that no clustering, K-means clustering, and hierarchical clustering patterns of data were correctly identified by the methods. It has also been observed that pcLasso method of regularization perform best across the different clustering methods while leaving cMCP behind as close competitor in all the categories of observed performance metrics such as deviance, misclassification error rate, number of significant coefficients, and AUC.

There are several avenues that one could follow for further research:

- Looking at the top right panels of Figures 1 and 3, we can see that the clusters are not well-separated at all. Despite this, the gap statistic split the predictors into a rather large number of clusters, especially for the leukemia data set, which may be a result of the large number of predictors. Perhaps another measure to determine the optimal number of clusters can be used. It is also worth mentioning that the computation time when using the `clusGap` function was on the magnitude of *hours*, making it non-practical for larger data sets.

- The two clustering algorithms used were chosen because of their simplicity and reputation, but one could argue that their poor performance is entirely expected. Much work has been done in the field of unsupervised learning, and one could use a different clustering algorithm to generate the group structure.

- There are many more grouped regularization models that can be employed that perform bi-variate selection, such as the group exponential lasso (“GEL”) (Breheny, 2015) or the group bridge (Huang et al., 2009), that can be efficiently implemented using the `grpreg` package. Each of these methods have their own interesting properties, and one could investigate if their use can improve over the non-clustered models.

Acknowledgements

We would like to thank J. Kenneth Tay for his helpful comments about pcLasso and undergraduate research in general. We would also like to thank Patrick Breheny for his helpful comments about his `grpreg` package. The authors were supported by NSF Award 1757717.

References

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999), "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences*, 96, 6745–6750.

Breheny, P. and Huang, J. (2009), "Penalized methods for bi-level variable selection," *Statistics and its interface*, 2, 369.

— (2015), "Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors," *Statistics and Computing*, 25, 173–187.

Breheny, P. J. (2015), "The group exponential lasso for bi-level variable selection." *Biometrics*, 71 3, 731–40.

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, 33, 1.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al. (1999), "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, 286, 531–537.

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

Huang, J., Breheny, P., and Ma, S. (2012), "A selective review of group selection in high-dimensional models," *Statistical Science: a Review Journal of the Institute of Mathematical Statistics*, 27.

Huang, J., Ma, S., Xie, H., and Zhang, C.-H. (2009), "A group bridge approach for variable selection," *Biometrika*, 96, 339–355.

Lee, J., Ciccarello, S., Acharjee, M., and Das, K. (2018), "Dimension reduction of gene expression data," *Journal of Statistical Theory and Practice*, 12, 450–461.

MacQueen, J. et al. (1967), "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Oakland, CA, USA, vol. 1, pp. 281–297.

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

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013), "A sparse-group lasso," *Journal of Computational and Graphical Statistics*, 22, 231–245.

Simon, N. and Tibshirani, R. (2012), "Standardization and the group lasso penalty," *Statistica Sinica*, 22, 983.

Tay, J. K., Friedman, J., and Tibshirani, R. (2018), "Principal component-guided sparse regression," *arXiv e-prints*, arXiv:1810.04651.

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

Tibshirani, R., Walther, G., and Hastie, T. (2001), "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63, 411–423.

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

Zhang, C. H. (2007), "Penalized linear unbiased selection," *Department of Statistics and Bioinformatics, Rutgers University*, 3.

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

Received: *3 September 2020*

Accepted: *1 March 2022*