glmnetcr: An R Package for Ordinal Response Prediction in High-dimensional Data Settings

Kellie J. Archer
Virginia Commonwealth University

Abstract

This paper describes an R package, **glmnetcr**, that provides a function for fitting a penalized continuation ratio model when interest lies in predicting an ordinal response. The function, `glmnet.cr` uses the coordinate descent fitting algorithm as implemented in `glmnet` and described by (Friedman, Hastie, and Tibshirani 2010). Methods for extracting all estimated coefficients, extracting non-zero coefficient estimates, obtaining the predicted class, and obtaining the class-specific fitted probabilities have been implemented. Additionally, generic methods from **glmnet** including `print` and `plot` can be applied to a `glmnet.cr` object.

Keywords: ordinal response, penalized models, LASSO, L_1 constraint, R.

1. Introduction

High-throughput genomic experiments are frequently conducted for the purpose of examining whether genes are predictive of or significantly associated with phenotype. In many biomedical settings where histopathological or health status data are collected, phenotypic variables are recorded on an ordinal scale. Nevertheless, most often investigators neglect the ordinality of the phenotypic data and rather dichotomize the ordinal class then apply statistical methods suitable for two-class comparisons and predictions. This tendency to analyze ordinal data using dichotomous class methodologies may be due to the lack of available statistical methods and software for modeling an ordinal response in the presence of a high-dimensional covariate space. The approach of collapsing ordinal categories may neglect important information in the study (Armstrong and Sloan 1989).

A variety of statistical modeling procedures, namely, proportional odds, adjacent category, stereotype logit, and continuation ratio models can be used to predict an ordinal response. In this paper, we focus attention to the continuation ratio model because its likelihood can be easily re-expressed such that existing software can be readily used for model fitting. The backward formulation of the continuation ratio models the logit as

$$\texttt{logit}\left(\frac{P(Y=k|X=x)}{P(Y\leq k|X=x)}\right)=\alpha_k+\boldsymbol{\beta}_k^T\mathbf{x} \tag{1}$$

whereas the forward formulation models the logit as

$$\texttt{logit}\left(\frac{P(Y=k|X=x)}{P(Y\geq k|X=x)}\right)=\alpha_k+\boldsymbol{\beta}_k^T\mathbf{x}. \tag{2}$$

Rather than describe both formulations in detail, here we present the backward formulation, which is commonly used when progression through disease states from none, mild, moderate, severe is represented by increasing integer values, and interest lies in estimating the odds of more severe disease compared to less severe disease (Bender and Benner 2000). Suppose each observation, $i = 1, \ldots, n$, belongs to one ordinal class $k = 1, \ldots, K$. Therefore for $i = 1, \ldots, n$ we can construct a vector $\mathbf{y}_i$ to represent ordinal class membership, such that $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{iK})^T$, where $y_{ik} = 1$ if the response is in category k and 0 otherwise, so that $n_i = \sum_{k=1}^{K} y_{ik} = 1$. Using the logit link, the equation representing the conditional probability for class k is

$$\delta_k(x) = P(Y = k | Y \leq k, X = x) = \frac{\exp(\alpha_k + \boldsymbol{\beta}^T \mathbf{X})}{1 + \exp(\alpha_k + \boldsymbol{\beta}^T \mathbf{X})}. \tag{3}$$

The likelihood for the continuation ratio model is then the product of conditionally independent binomial terms (Cox 1975), which is given by

$$L(\boldsymbol{\beta}|\mathbf{y}, \mathbf{x}) = \prod_{i=1}^{n} \delta_2^{y_{i2}} (1 - \delta_2)^{n_i - \sum_{k=2}^{K} y_{ik}} \delta_3^{y_{i3}} (1 - \delta_3)^{n_i - \sum_{k=3}^{K} y_{ik}} \times \cdots \times \delta_K^{y_{iK}} (1 - \delta_K)^{n_i - y_{iK}} \tag{4}$$

where here we have simplified our notation by not explicitly including the dependence of the conditional probability δ_k on $\mathbf{x}$. Further, simplifying our notation to let $\boldsymbol{\beta}$ represent the vector containing both the thresholds $(\alpha_2, \ldots, \alpha_K)$ and the log odds $(\beta_1, \ldots, \beta_p)$ for all $K - 1$ logits, the full parameter vector is

$$\boldsymbol{\beta} = (\alpha_2, \beta_{21}, \beta_{22}, \ldots, \beta_{2p}, \alpha_3, \beta_{31}, \beta_{32}, \ldots, \beta_{3p}, \alpha_K, \beta_{K,1}, \beta_{K,2}, \ldots, \beta_{K,p})^T \tag{5}$$

which is of length $(K-1)(p+1)$. As can be seen from equation 4, the likelihood can be factored into $K - 1$ independent likelihoods, so that maximization of the independent likelihoods will lead to an overall maximum likelihood estimate for all terms in the model (Bender and Benner 2000). A model consisting of $K-1$ different β vectors may be overparameterized so to simplify, one commonly fits a constrained continuation model, which includes the $K - 1$ thresholds $(\alpha_2, \ldots, \alpha_K)$ and one common set of p slope parameters, $(\beta_1, \ldots, \beta_p)$. To fit a constrained continuation ratio model, the original dataset can be restructured by forming $K - 1$ subsets, where for classes $k = 2, \ldots, K$, the subset contains those observations in the original dataset up to class k. Additionally, for the k^{th} subset, the outcome is dichotomized as $y = 1$ if the ordinal class is k and $y = 0$ otherwise. Furthermore, an indicator is constructed for each subset representing subset membership. Thereafter the $K - 1$ subsets are appended to form the restructured dataset, which represents the $K - 1$ conditionally independent datasets in equation 4. Applying a logistic regression model to this restructured dataset yields an L_1 constrained continuation ratio model.

2. Penalized Models

For datasets where the number of covariates p exceeds the sample size n, the backwards stepwise procedure cannot be undertaken. Furthermore, for any problem using a forward selection procedure the discrete variable inclusion process can exhibit high variance. Moreover, for high-dimensional covariate spaces, the best subset procedure is computationally prohibitive. Two penalized methods, ridge and L_1 penalization, places a penalty on a function of the coefficient

estimates, thereby permitting a model fit even for high-dimensional data Tibshirani (1996, 1997). A generalization of these penalized models can be expressed as,

$$\tilde{\beta} = \arg\min_{\beta}(\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|^q) \tag{6}$$

for $q \geq 0$. When $q = 1$ we have the an L_1 penalized model, when $q = 2$ we have ridge regression. Values of $q \in (1, 2)$ provide a compromise between the L_1 and ridge penalized models. Because when $q > 1$ coefficients are no longer set exactly equal to 0, the elastic net penalty was introduced

$$\lambda\sum_{j=1}^{p}(\alpha\beta_j^2 + (1-\alpha)|\beta_j|). \tag{7}$$

3. Implementation

The **glmnetcr** package was written in the R programming environment (R Development Core Team 2009) and depends on the **glmnet** package (Park and Hastie 2007). Similar to the **Design** package which includes a function `cr.setup` for restructuring a dataset for fitting a forward continuation ratio model, in this package the model is fit by restructuring the dataset then passing the restructured dataset to a penalized logistic regression fitting function. However, unlike `cr.setup` which produces an object of class `list` from which the response and restructured independent variables are extracted and passed to a model fitting algorithm, in the **glmnetcr** package the restructuring functions are transparent to the user. Specifically, the **glmnetcr** package fits either a forward or backward (default) penalized constrained continuation ratio model by specification of `method="forward"` in the `glmnet.cr` call. The `glmnet.cr` function restructures the dataset to represent the $K-1$ conditionally independent likelihoods and then fits the penalized continuation ratio model using the **glmnet** framework. Thereafter, the coordinate descent fitting procedure used by the `glmnet` function in the **glmnet** package is used in fitting the penalized continuation ratio model when invoking `glmnet.cr`. This allows fitting a penalized model for situations where the number of covariates p exceed the sample size n. In addition, methods for extracting the best fitting model from the path using AIC and BIC criteria, obtaining predicted class and fitted class probabilities, and returning coefficient estimates were written in addition to the `print` and `plot` methods copied from **glmnet**.

4. Example

The **glmnetcr** package includes a filtered microarray dataset `diabetes` in which asymptomatic males not previously diagnosed with Type II diabetes were enrolled and subsequently were cross-classified as either normal controls (N=8), having impaired fasting glucose (N=7), or as Type II diabetics (N=9) based on a fasting glucose intolerance test. From the code below we can see that the classification variable is stored as `y` in the first column of the `diabetes data.frame`; all subsequent columns are the 11,066 Illumina probes having no negative expression values. In fitting the model we can extract the covariates into an object `x` and the ordinal outcome into the object `y`. The code for fitting a backward (default) continuation ratio model is given by

```
> library(glmnetcr)

Loaded glmnet 1.5.1

> data(diabetes)
> dim(diabetes)

[1]    24 11067

> names(diabetes)[1:10]

 [1] "y"            "ILMN_1343291" "ILMN_1651228" "ILMN_1651229" "ILMN_1651236"
 [6] "ILMN_1651254" "ILMN_1651262" "ILMN_1651268" "ILMN_1651278" "ILMN_1651286"

> summary(diabetes$y)

                control impaired fasting glucose          type 2 diabetes
                      8                       7                        9

> x <- diabetes[, 2:dim(diabetes)[2]]
> y <- diabetes$y
> fit <- glmnet.cr(x, y)
```

As with `glmnet` model objects, methods such as `print` and `plot` can be applied to `glmnet.cr` model objects, which are helpful for selecting the step at which to select the final model from the solution path.

```
> print(fit)

Call:  glmnet(x = glmnet.data$x, y = glmnet.data$y, family = "binomial",      weights = we

      Df     %Dev   Lambda
 [1,]  2 0.006051 0.391400
 [2,]  3 0.054150 0.373600
 [3,]  3 0.098280 0.356600
 [4,]  3 0.139000 0.340400
 [5,]  3 0.176600 0.324900
 [6,]  3 0.211600 0.310200
 [7,]  3 0.244200 0.296100
 [8,]  3 0.274800 0.282600
 [9,]  3 0.303400 0.269800
[10,]  3 0.330400 0.257500
[11,]  3 0.355800 0.245800
[12,]  3 0.379800 0.234600
[13,]  3 0.402600 0.224000
[14,]  3 0.424200 0.213800
```

```
[15,]  3 0.444800 0.204100
[16,]  3 0.464300 0.194800
[17,]  3 0.483000 0.185900
[18,]  3 0.500800 0.177500
[19,]  3 0.517900 0.169400
[20,]  3 0.534200 0.161700
[21,]  4 0.551100 0.154400
[22,]  4 0.568700 0.147400
[23,]  4 0.585600 0.140700
[24,]  4 0.601800 0.134300
[25,]  5 0.617800 0.128200
[26,]  6 0.634100 0.122300
[27,]  6 0.649700 0.116800
[28,]  6 0.664700 0.111500
[29,]  6 0.679100 0.106400
[30,]  6 0.692900 0.101600
[31,]  7 0.706400 0.096950
[32,]  7 0.719300 0.092550
[33,]  7 0.731700 0.088340
[34,]  7 0.743600 0.084320
[35,]  7 0.755000 0.080490
[36,]  7 0.765900 0.076830
[37,]  7 0.776400 0.073340
[38,]  7 0.786400 0.070010
[39,]  7 0.796000 0.066830
[40,]  7 0.805100 0.063790
[41,]  7 0.813900 0.060890
[42,]  7 0.822200 0.058120
[43,]  7 0.830200 0.055480
[44,]  7 0.837900 0.052960
[45,]  7 0.845200 0.050550
[46,]  8 0.852200 0.048250
[47,]  9 0.859000 0.046060
[48,] 11 0.865500 0.043970
[49,] 12 0.871800 0.041970
[50,] 12 0.877800 0.040060
[51,] 12 0.883500 0.038240
[52,] 12 0.888900 0.036500
[53,] 12 0.894100 0.034840
[54,] 12 0.899000 0.033260
[55,] 12 0.903700 0.031750
[56,] 12 0.908200 0.030300
[57,] 12 0.912400 0.028930
[58,] 12 0.916500 0.027610
[59,] 12 0.920300 0.026360
[60,] 12 0.924000 0.025160
[61,] 12 0.927500 0.024020
```

```
 [62,] 13 0.930800 0.022920
 [63,] 13 0.934000 0.021880
 [64,] 13 0.937000 0.020890
 [65,] 14 0.939900 0.019940
 [66,] 14 0.942700 0.019030
 [67,] 14 0.945300 0.018170
 [68,] 14 0.947800 0.017340
 [69,] 14 0.950200 0.016550
 [70,] 15 0.952500 0.015800
 [71,] 15 0.954700 0.015080
 [72,] 15 0.956800 0.014400
 [73,] 15 0.958700 0.013740
 [74,] 15 0.960600 0.013120
 [75,] 15 0.962400 0.012520
 [76,] 15 0.964200 0.011950
 [77,] 15 0.965800 0.011410
 [78,] 17 0.967400 0.010890
 [79,] 17 0.968800 0.010400
 [80,] 17 0.970300 0.009923
 [81,] 17 0.971600 0.009472
 [82,] 17 0.972900 0.009042
 [83,] 17 0.974200 0.008631
 [84,] 17 0.975300 0.008239
 [85,] 17 0.976500 0.007864
 [86,] 17 0.977500 0.007507
 [87,] 17 0.978500 0.007165
 [88,] 17 0.979500 0.006840
 [89,] 17 0.980400 0.006529
 [90,] 17 0.981300 0.006232
 [91,] 17 0.982200 0.005949
 [92,] 17 0.983000 0.005679
 [93,] 17 0.983800 0.005420
 [94,] 17 0.984500 0.005174
 [95,] 17 0.985200 0.004939
 [96,] 17 0.985900 0.004714
 [97,] 17 0.986500 0.004500
 [98,] 17 0.987100 0.004296
 [99,] 17 0.987700 0.004100
[100,] 17 0.988200 0.003914
```

```
> plot(fit, xvar = "step", type = "bic")
> plot(fit, xvar = "step", type = "coefficients")
```

Note that when plotting, the horizontal axis can be `"norm"`, `"lambda"`, or `"step"`, however extractor functions for `glmnet.cr` generally require the step to be selected, so we have selected `xvar = "step"` in this example. The vertical axis can be `"coefficients"`, `"aic"`, or `"bic"`. As one can see, there is a multitude of models fit from one call to `glmnet.cr`. To facil-

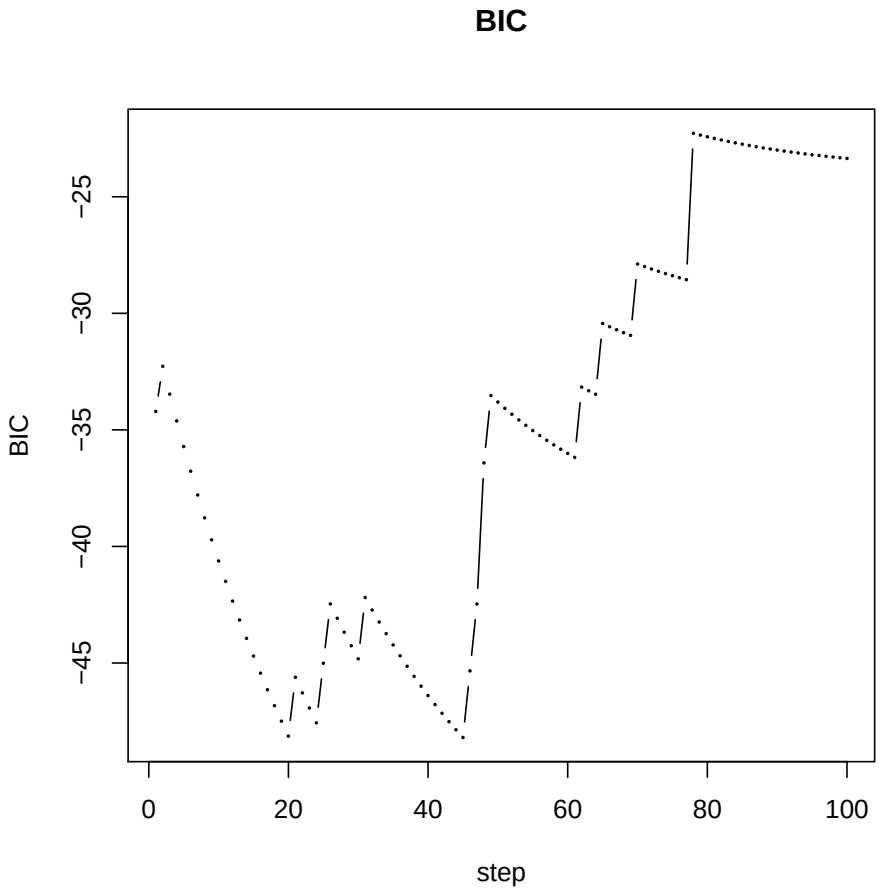

Figure 1: Plot of Bayesian Information Criteria across the regularization path for the fitted `glmnet.cr` object using the `diabetes` data.

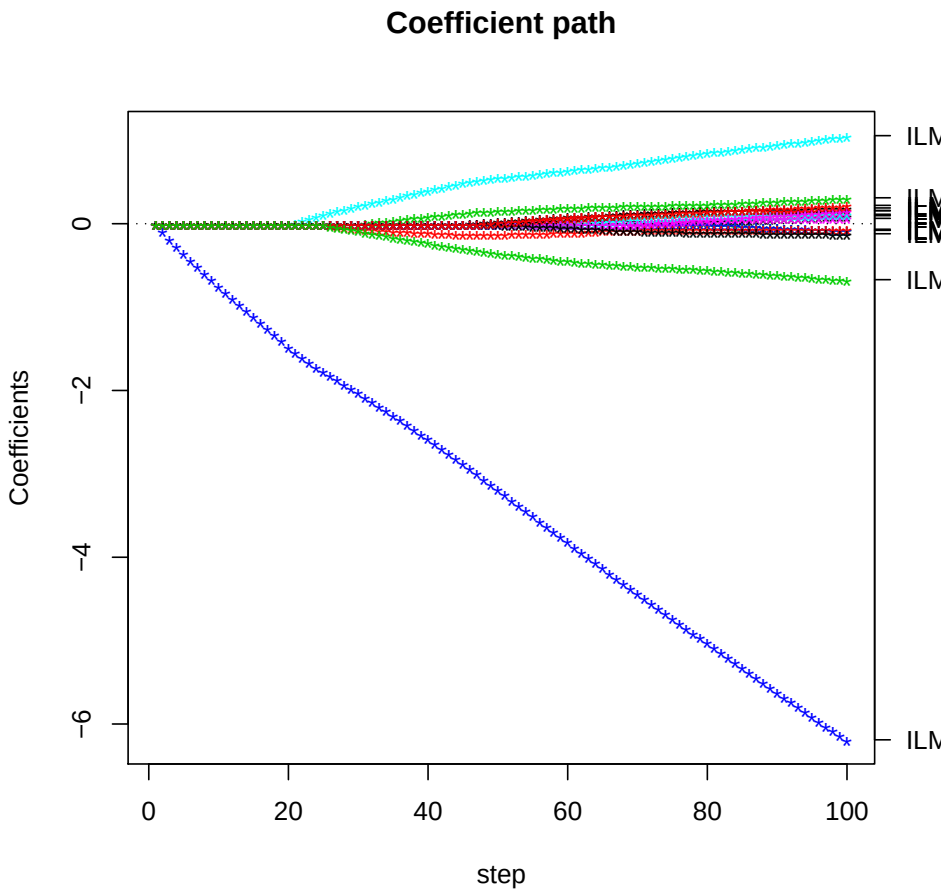

Figure 2: Plot of estimated coefficients across the regularization path for the fitted `glmnet.cr` object using the `diabetes` data.

iate extraction of best fitting models using commonly used criterion, the `select.glmnet.cr` function can be used. The `select.glmnet.cr` function extracts the best fitting model from the solution path, where the `which` parameter allows one to select either AIC or by default, BIC.

```
> BIC.step <- select.glmnet.cr(fit)
> BIC.step

s44
 45

> AIC.step <- select.glmnet.cr(fit, which = "AIC")
> AIC.step

s44
 45
```

In this example, although Step 45 corresponds to the model attaining the minimum BIC, it is clear from the plot that Step 20 results in a virtually equivalent BIC and is more parsimonious. Therefore we encourage use of plots in conjunction with the `select.glmnet.fit` function for identifying the model to be extracted.

A forward continuation ratio model can be fit using the syntax

```
> fit <- glmnet.cr(x, y, method = "forward")
```

Again, the `select.glmnet.cr` function extracts the best fitting model from the solution path, where the `which` parameter allows one to select either AIC or by default, BIC.

```
> BIC.step <- select.glmnet.cr(fit)
> BIC.step

s20
 21
```

The `coef` function returns all estimated coefficients for a `glmnet.cr` fitted model, including the intercept which is returned as `a0` as well as the estimated slope and threshold estimates stored in `beta`. The coefficient estimates are returned for a specific step of the regularization path by specifying the parameter `s`. The `nonzero.glmnet.cr` function returns only those non-zero coefficient estimates for a selected model.

```
> coefficients <- coef(fit, s = BIC.step)
> coefficients$a0

      s20
-3.494128

> sum(coefficients$beta != 0)
```

```
[1] 3

> nonzero.glmnet.cr(fit, s = BIC.step)

$a0
      s20
-3.494128

$beta
ILMN_1759232          cp1          cp2
  0.01877673  -1.53138530   0.18375244
```

Note that the `glmnet.cr` function fits a penalized constrained continuation ratio model; therefore for K classes, there will be $K-1$ intercepts representing the cutpoints between adjacent classes. In this package, the nomenclature for these cutpoints is to use "cp$k$" where $k = 1, \ldots, K-1$. In this dataset, $K = 3$ so the intercepts are `cp1` and `cp2` with `a0` being an offset. When using the BIC (Step 21) to select the final model, the only probe having a nonzero coefficient estimate `ILMN_1759232` which corresponds to the insulin receptor substrate 1 (IRS1) gene which is biologically meaningful.

Continuation ratio models predicts conditional probabilities so a new method to extract the fitted probabilities and predicted class was created. The `predict` returns the AIC, BIC, predicted class, and the fitted probabilities for the K classes for all steps along the regularization path. By default the training data is used to obtain model predictions, though predicted class and fitted probabilities can be obtained for a test dataset by specifying a different dataset using the `newx` parameter. The `fitted` function extracts the AIC, BIC, predicted class, and the fitted probabilities for the K classes for a specific step of the regularization path by specifying the parameter `s`.

```
> hat <- fitted(fit, s = BIC.step)
> names(hat)

[1] "BIC"   "AIC"   "class" "probs"

> table(hat$class, y)

                          y
                           control impaired fasting glucose type 2 diabetes
  control                        8                       1                0
  impaired fasting glucose       0                       6                0
  type 2 diabetes                0                       0                9
```

Summary

Herein we have described the **glmnetcr** package which works in conjunction with the **glmnet** package in the R programming environment. The package provides methods for fitting

either a forward or backward penalized continuation ratio model. Moreover, the likelihood-based penalized constrained continuation ratios models have been demonstrated to have good performance in simulation studies and when applied to microarray gene expression datasets (Archer and Williams 2010), as well as in comparison to penalized Bayesian continuation ratio models using their encoded sparsity priors (Kiiveri 2008). Therefore the **glmnetcr** package should be helpful when predicting an ordinal response for datasets where the number of covariates exceeds the number of available samples.

Acknowledgments

This research was supported by the National Institute of Library Medicine R03LM009347.

References

Archer KJ, Williams AA (2010). "L_1 penalized continuation ratio models for ordinal response prediction using high-dimensional datasets." *Statistics in Medicine*, **under review**.

Armstrong B, Sloan M (1989). "Ordinal regression models for epidemiologic data." *American Journal of Epidemiology*, **129**, 191–204.

Bender R, Benner A (2000). "Calculating ordinal regression models in SAS and S-Plus." *Biometrical Journal*, **42**, 677–699.

Cox D (1975). "Partial likelihood." *Biometrika*, **62**, 269–276.

Friedman J, Hastie T, Tibshirani R (2010). "Regularization paths for generalized linear models via coordinate descent." *Journal of Statistical Software*, **33**, 1–22.

Kiiveri HT (2008). "A general approach to simultaneous model fitting and variable elimination in response models for biological data with many more variables than observations." *BMC Bioinformatics*, **9**, 195.

Park MY, Hastie T (2007). *glmpath: L1 Regularization Path for Generalized Linear Models and Cox Proportional Hazards Model*. R package version 0.94.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

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

Tibshirani R (1997). "The lasso method for variable selection in the Cox model." *Statistics in Medicine*, **16**, 385–395.

Affiliation:

Kellie J. Archer
Department of Biostatistics

Virginia Commonwealth University
Box 980032
Richmond, VA 23298-0032
E-mail: kjarcher@vcu.edu
URL: http://www.people.vcu.edu/~kjarcher/