glmpathcr: 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, **glmpathcr**, that provides a function for fitting a penalized continuation ratio model when interest lies in predicting an ordinal response. The function, `glmpath.cr` uses the coordinate descent fitting algorithm as implemented in `glmpath` and described by (Park and Hastie 2007a). 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 **glmpath** including `summary`, `print`, and `plot` can be applied to a `glmpath.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

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

whereas the forward formulation models the logit as

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

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, $i = 1, \dots, n$, belongs to one ordinal class $k = 1, \dots, K$. Therefore for $i = 1, \dots, n$ we can construct a vector $\mathbf{y}_i$ to represent ordinal class membership, such that $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{iK})^T$, 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

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

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

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

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 $(\alpha_2, \dots, \alpha_K)$ and the log odds $(\beta_1, \dots, \beta_p)$ for all $K-1$ logits, the full parameter vector is

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

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).

2. Implementation

The **glmpathcr** package was written in the R programming environment (R Development Core Team 2009) and depends on the **glmpath** package (Park and Hastie 2007b). 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 **glmpathcr** package the restructuring functions are transparent to the user. Specifically, the **glmpathcr** package fits either a forward or backward (default) penalized constrained continuation ratio model by specification of `method="forward"` in the `glmpath.cr` call. The `glmpath.cr` function restructures the dataset to represent the $K-1$ conditionally independent likelihoods and then fits the penalized continuation ratio model using the **glmpath** framework. Therefore, the coordinate descent fitting procedure used by the `glmpath` function in the **glmpath** package are used in fitting the penalized continuation ratio model when invoking `glmpath.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`, `summary`, and `plot` methods copied from **glmpath**.

3. Example

A simulated dataset, `data`, consisting of 1,000 covariates and a three-class ordinal response with 30 observations in each class is included in the **glmpath** package for testing ordinal classification methodologies. The first column (`V1`) is stochastically associated with the ordinal response: for class 0, `V1` is distributed as N(0,1); for class 1, `V1` is distributed as N(1.5,1); and for class 2, `V1` is distributed as N(3,1). All other predictor variables (`V2`-`V1000`) are multivariable normally distributed with mean vector **0** and variance-covariance matrix **I**. Therefore the Bayes Error associated with this dataset is 0.302. The last column in `data` is the ordinal response, `class`. The code for fitting a backward (default) continuation ratio model is given by

```
> library(glmpathcr)
> data(data)
> x <- data[, 1:1000]
> y <- data$class
> fit <- glmpath.cr(x, y)
```

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

```
> summary(fit)

          Df     Deviance      AIC      BIC
Step 1     3 1.977502e+02 203.7502 212.7821
Step 7     4 1.268914e+02 134.8914 146.9339
Step 9     6 1.236720e+02 135.6720 153.7358
Step 10    7 1.235747e+02 137.5747 158.6492
Step 12    8 1.229605e+02 138.9605 163.0455
Step 13    9 1.213437e+02 139.3437 166.4394
Step 15   10 1.206910e+02 140.6910 170.7974
Step 16   11 1.206072e+02 142.6072 175.7241
Step 18   12 1.185045e+02 142.5045 178.6321
Step 19   13 1.140777e+02 140.0777 179.2160
Step 21   14 1.059049e+02 133.9049 176.0538
Step 23   15 1.052149e+02 135.2149 180.3744
Step 25   16 1.025033e+02 134.5033 182.6735
Step 27   17 9.788337e+01 131.8834 183.0642
Step 29   18 9.250006e+01 128.5001 182.6915
Step 31   19 8.937950e+01 127.3795 184.5816
Step 32   20 8.936458e+01 129.3646 189.5773
Step 34   21 8.913574e+01 131.1357 194.3591
```

```
Step 35    22 8.740808e+01 131.4081 197.6421
Step 37    23 8.391957e+01 129.9196 199.1642
Step 39    24 8.211261e+01 130.1126 202.3679
Step 41    25 7.883779e+01 128.8378 204.1037
Step 43    26 7.744341e+01 129.4434 207.7199
Step 45    27 7.552956e+01 129.5296 210.8167
Step 47    28 7.464407e+01 130.6441 214.9419
Step 49    29 7.331620e+01 131.3162 218.6246
Step 50    30 7.128100e+01 131.2810 221.6001
Step 52    31 7.032096e+01 132.3210 225.6506
Step 54    32 6.973133e+01 133.7313 230.0717
Step 56    33 6.955485e+01 135.5548 234.9058
Step 58    34 6.941267e+01 137.4127 239.7743
Step 60    35 6.655017e+01 136.5502 241.9224
Step 62    36 6.591471e+01 137.9147 246.2976
Step 64    37 6.396196e+01 137.9620 249.3555
Step 67    38 5.591195e+01 131.9120 246.3161
Step 70    39 4.964509e+01 127.6451 245.0599
Step 73    40 4.588916e+01 125.8892 246.3146
Step 75    41 4.518095e+01 127.1809 250.6170
Step 77    42 4.439213e+01 128.3921 254.8388
Step 80    43 4.097572e+01 126.9757 256.4330
Step 82    44 3.855674e+01 126.5567 259.0247
Step 84    45 3.771496e+01 127.7150 263.1935
Step 86    46 3.683175e+01 128.8317 267.3210
Step 88    47 3.561162e+01 129.6116 271.1115
Step 90    48 3.533279e+01 131.3328 275.8433
Step 91    48 3.389858e+01 129.8986 274.4091
Step 93    48 3.248141e+01 128.4814 272.9919
Step 95    49 3.148533e+01 129.4853 277.0065
Step 96    49 3.115469e+01 129.1547 276.6758
Step 98    49 2.906390e+01 127.0639 274.5850
Step 101   50 2.708599e+01 127.0860 277.6178
Step 103   51 2.597671e+01 127.9767 281.5191
Step 105   52 2.523697e+01 129.2370 285.7900
Step 107   53 2.454167e+01 130.5417 290.1053
Step 110   54 2.252000e+01 130.5200 293.0943
Step 111   55 2.248267e+01 132.4827 298.0676
Step 114   56 1.674256e+01 128.7426 297.3381
Step 116   57 1.643027e+01 130.4303 302.0365
Step 119   58 1.221607e+01 128.2161 302.8329
Step 121   59 1.160682e+01 129.6068 307.2343
Step 122   59 1.048302e+01 128.4830 306.1105
Step 124   59 1.036497e+01 128.3650 305.9925
Step 126   60 9.736097e+00 129.7361 310.3742
Step 129   61 8.882654e+00 130.8827 314.5314
Step 130   61 8.367347e+00 130.3673 314.0161
```

```
Step 133  61 7.613208e+00 129.6132 313.2620
Step 136  62 6.748127e+00 130.7481 317.4075
Step 139  63 5.756419e+00 131.7564 321.4264
Step 141  64 5.720197e+00 133.7202 326.4009
Step 144  65 4.672943e+00 134.6729 330.3642
Step 147  66 4.320249e+00 136.3202 335.0222
Step 152  67 2.735014e+00 136.7350 338.4476
Step 155  68 1.247403e+00 137.2474 341.9706
Step 158  69 9.716167e-01 138.9716 346.7055
Step 162  70 5.010593e-01 140.5011 351.2455
Step 165  71 3.818247e-01 142.3818 356.1369
Step 166  70 3.476473e-01 140.3476 351.0921
Step 170  70 8.522756e-02 140.0852 350.8297
Step 173  71 6.081049e-02 142.0608 355.8159
Step 174  71 3.708819e-02 142.0371 355.7922
Step 176  71 3.342187e-02 142.0334 355.7885
Step 177  71 2.938603e-02 142.0294 355.7845
Step 179  71 2.541967e-02 142.0254 355.7805
Step 180  72 1.844144e-02 144.0184 360.7842
Step 182  73 1.449531e-02 146.0145 365.7909
Step 183  74 1.278486e-02 148.0128 370.7998
Step 184  74 1.264285e-02 148.0126 370.7997
Step 185  74 1.207392e-02 148.0121 370.7991
Step 186  74 1.180871e-02 148.0118 370.7988
Step 187  74 1.160685e-02 148.0116 370.7986
Step 188  74 8.075618e-03 148.0081 370.7951
Step 189  74 7.642540e-03 148.0076 370.7947
Step 190  74 7.474924e-03 148.0075 370.7945
Step 191  73 7.421659e-03 146.0074 365.7838
Step 192  73 6.660177e-03 146.0067 365.7830
Step 193  74 6.538656e-03 148.0065 370.7936
Step 194  75 6.202576e-03 150.0062 375.8038
Step 195  76 6.109357e-03 152.0061 380.8144
Step 196  78 5.805791e-03 156.0058 390.8354
Step 197  78 5.801257e-03 156.0058 390.8354
Step 198  77 5.601975e-03 154.0056 385.8245
Step 199  77 5.504980e-03 154.0055 385.8244
Step 200  77 5.245695e-03 154.0052 385.8242
Step 202  77 4.752132e-03 154.0048 385.8237
Step 203  78 4.656521e-03 156.0047 390.8342
Step 204  79 4.530401e-03 158.0045 395.8447
Step 206  80 4.305395e-03 160.0043 400.8551
Step 207  80 4.293758e-03 160.0043 400.8551
Step 209  80 3.787662e-03 160.0038 400.8546
Step 211  81 3.650458e-03 162.0037 405.8651
Step 212  82 3.453881e-03 164.0035 410.8755
Step 213  82 3.423946e-03 164.0034 410.8755
```

```
Step 214  81 3.383232e-03 162.0034 405.8648
Step 215  82 3.274787e-03 164.0033 410.8754
Step 216  82 3.013102e-03 164.0030 410.8751
Step 217  82 2.942978e-03 164.0029 410.8750
Step 218  84 2.832763e-03 168.0028 420.8962
Step 219  84 2.771885e-03 168.0028 420.8961
Step 220  85 2.697916e-03 170.0027 425.9067
Step 222  86 2.487565e-03 172.0025 430.9171
Step 223  87 2.437891e-03 174.0024 435.9277
Step 224  89 2.375010e-03 178.0024 445.9489
Step 225  90 2.328724e-03 180.0023 450.9595
Step 226  91 2.279899e-03 182.0023 455.9701
Step 227  90 2.245480e-03 180.0022 450.9594
Step 228  90 2.151490e-03 180.0022 450.9593
Step 229  92 2.121618e-03 184.0021 460.9806
Step 230  93 2.093159e-03 186.0021 465.9912
Step 231  93 2.062546e-03 186.0021 465.9911
Step 232  92 2.029814e-03 184.0020 460.9805
Step 233  91 1.970488e-03 182.0020 455.9698
Step 234  91 1.919379e-03 182.0019 455.9697
Step 235  92 1.896475e-03 184.0019 460.9803
Step 236  93 1.828767e-03 186.0018 465.9909
Step 237  93 1.825758e-03 186.0018 465.9909
Step 238  92 1.821194e-03 184.0018 460.9803
Step 239  91 1.808816e-03 182.0018 455.9696
Step 240  90 1.762903e-03 180.0018 450.9589
Step 241  91 1.709718e-03 182.0017 455.9695
Step 242  93 1.642881e-03 186.0016 465.9907
Step 243  95 1.604030e-03 190.0016 476.0120
Step 244  97 1.575597e-03 194.0016 486.0332
Step 245  99 1.531090e-03 198.0015 496.0544
Step 246 100 1.488077e-03 200.0015 501.0650
Step 247 100 1.465819e-03 200.0015 501.0650
Step 248 100 1.409207e-03 200.0014 501.0649
Step 249 103 1.369562e-03 206.0014 516.0968
Step 250 106 1.332009e-03 212.0013 531.1287
Step 251 106 1.316473e-03 212.0013 531.1287
Step 252 106 1.312051e-03 212.0013 531.1287
Step 253 105 1.302143e-03 210.0013 526.1180
Step 254 104 1.298270e-03 208.0013 521.1074
Step 255 103 1.277893e-03 206.0013 516.0967
Step 256 104 1.243826e-03 208.0012 521.1073
Step 257 106 1.224708e-03 212.0012 531.1286
Step 258 106 1.211353e-03 212.0012 531.1286
Step 259 108 1.189521e-03 216.0012 541.1498
Step 260 109 1.149689e-03 218.0011 546.1604
Step 261 111 1.117974e-03 222.0011 556.1816
```

```
Step 262 112 1.107710e-03 224.0011 561.1923
Step 263 112 1.077822e-03 224.0011 561.1922
Step 264 113 1.067414e-03 226.0011 566.2029
Step 265 118 1.046297e-03 236.0010 591.2560
Step 266 119 1.040973e-03 238.0010 596.2666
Step 267 118 1.040083e-03 236.0010 591.2560
Step 268 118 1.021432e-03 236.0010 591.2560
Step 269 118 1.011200e-03 236.0010 591.2560
Step 271 117 9.713459e-04 234.0010 586.2453
Step 272 117 9.558112e-04 234.0010 586.2453
Step 273 117 9.164918e-04 234.0009 586.2452
Step 274 121 8.957896e-04 242.0009 606.2878
Step 275 124 8.861962e-04 248.0009 621.3197
Step 276 128 8.771551e-04 256.0009 641.3622
Step 277 129 8.600954e-04 258.0009 646.3728
Step 278 130 8.366869e-04 260.0008 651.3834
Step 279 135 8.173980e-04 270.0008 676.4366
Step 280 136 8.038641e-04 272.0008 681.4472
Step 281 138 7.826633e-04 276.0008 691.4685
Step 282 140 7.259221e-04 280.0007 701.4897
Step 284 143 6.836433e-04 286.0007 716.5215
```

```
> plot(fit, xvar = "step", type = "bic", plot.all.steps = TRUE,
+     breaks = FALSE)
```

Note that when plotting, the horizontal axis can be `norm`, `lambda`, or `step`, however extractor functions for `glmpath.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 `glmpath.cr`. To faciliate extraction of best fitting models using commonly used criterion, the `model.select` function can be used. The `model.select` 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 <- model.select(fit)
> BIC.step
```

```
[1] 7
```

In this example, Step 7 corresponds to a 4 degree of freedom model having the minimum BIC of 146.9339.

The `coef` function returns all estimated coefficients for a `glmpath.cr` fitted model, where the model selected is indicated by step `s`. The `nonzero.coef` function returns only those non-zero coefficient estimates for a selected model.

```
> coefficients <- coef(fit, s = BIC.step)
> sum(coefficients != 0)
```

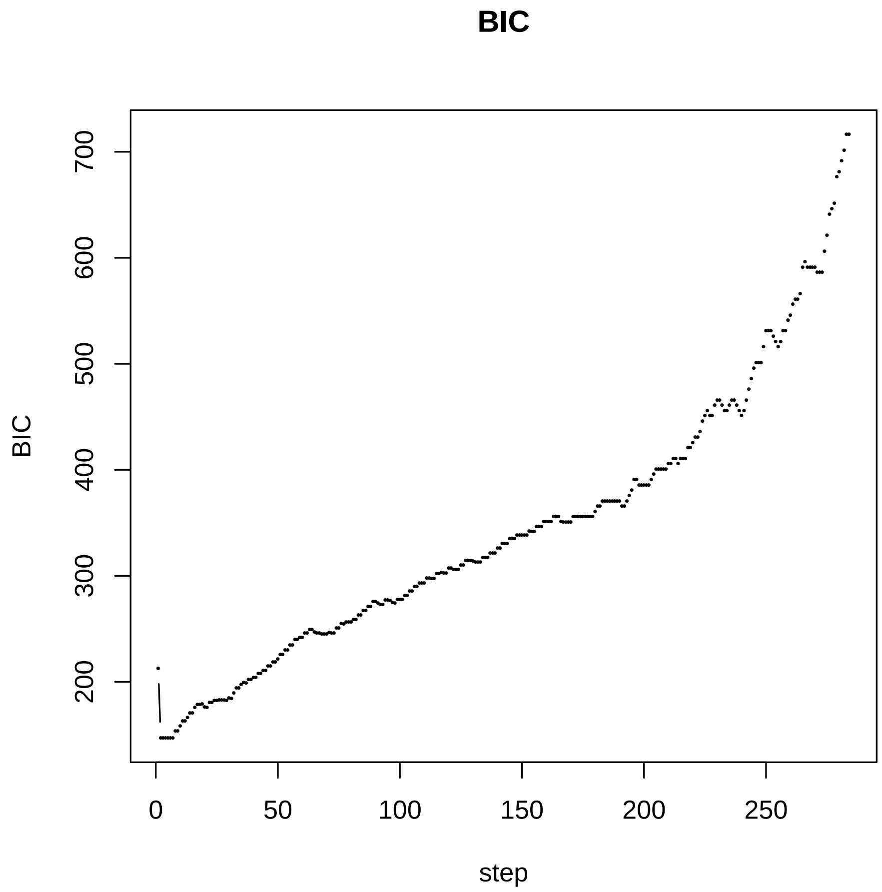

Figure 1: Plot of regularization path for `glmpath.cr` object using simulated dataset, `data`.

```
[1] 6

> nonzero.coef(fit, s = BIC.step)

    Intercept            V1          V285          V497           cp1
-1.3797425221  0.8460541902 -0.0040386156  0.0001098907  0.7940460195
          cp2
-0.7940460865
```

Note that the `glmpath.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 `Intercept` being an offset. When using the BIC to select the final model, the only variables having a non-zero coefficient estimate are the truly important covariate `V1` along with two noise covariates `V285` and `V497`.

Continuation ratio models predicts conditional probabilities so a new method to extract the fitted probabilities and predicted class was created. The `predict` and `fitted` functions are equivalent, and return either the predicted class or the fitted probabilities from the penalized continuation ratio model for a `glmpath.cr` object. The user is required to supply the fitted `glmpath.cr` model object, a data matrix `newx` that is either the same as the training data or an independent dataset having the same number and order of covariates as the training data, a vector `newy` that provides the class labels of the ordinal response. These functions extract the fitted values for the best fitting model using the BIC criteria by default, which can be changed to extracting the best fitting AIC model by supplying `which="AIC"`. By default, the predicted class is output. If one desired the fitted class-specific probabilities from the model, the `type="probs"` argument should be supplied.

```
> pred <- predict(fit, newx = x, newy = y)
> table(pred, y)

    y
pred  0  1  2
   1 25  7  0
   2  5 18  8
   3  0  5 22

> pred <- predict(fit, newx = x, newy = y, which = "AIC", type = "probs")
> pred[1:10, ]

             0          1            2
 [1,] 0.8824815 0.11378793 0.0037305240
 [2,] 0.4956481 0.47715531 0.0271965511
 [3,] 0.9737725 0.02546851 0.0007589597
 [4,] 0.9282863 0.06954289 0.0021708481
 [5,] 0.8861740 0.11022681 0.0035991901
```

```
 [6,] 0.9508358 0.04770922 0.0014550130
 [7,] 0.7375813 0.25257227 0.0098464754
 [8,] 0.7677757 0.22382923 0.0083950557
 [9,] 0.6868516 0.30059848 0.0125499351
[10,] 0.6447785 0.34013329 0.0150882334
```

Typically an unbiased estimate of error is desired. In this case, we can simulate a test dataset following the same procedure that was used to generate the original training set `data`. Afterward, we can apply the original model fit to the test set for estimating error. The `set.seed` function is used only to permit others to replicate these results.

```
> library(mvtnorm)
> set.seed(9)
> class1 <- rmvnorm(30, mean = rep(0, 1000), sigma = diag(1, nrow = 1000))
> class2 <- rmvnorm(30, mean = c(1.5, rep(0, 999)), sigma = diag(1,
+     nrow = 1000))
> class3 <- rmvnorm(30, mean = c(3, rep(0, 999)), sigma = diag(1,
+     nrow = 1000))
> class <- rep(0:2, each = 30)
> testset <- data.frame(cbind(rbind(class1, class2, class3), class))
> rm(class1, class2, class3, class)
> pred <- predict(fit, newx = testset[, 1:1000], newy = testset$class)
> table(pred, testset$class)
```

```
pred  0  1  2
   1 23 10  0
   2  6 13  9
   3  1  7 21
```

For illustrative purposes, a forward continuation ratio model can be fit using the syntax

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

and the predicted class can be obtained using

```
> pred <- predict(fit, newx = x, newy = y, method = "forward")
> table(pred, y)
```

```
    y
pred  0  1  2
   1 24  6  0
   2  6 19  7
   3  0  5 23
```

Summary

Herein we have described the **glmpathcr** package which works in conjunction with the **glmpath** package in the R programming environment. The package provides methods for fitting either a forward or backward penalized continuation ratio model. When applied to a simulated dataset having Bayes' error of 0.302, the method reported a test set error of 0.367. Moreover, the likelihood-based penalized continuation ratios models have been demonstrated to have good performance when applied to microarray gene expression datasets (Archer and Williams 2010) in comparison to corresponding penalized Bayesian continuation ratio models (Kiiveri 2008). Therefore the **glmpathcr** 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.

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 M, Hastie T (2007a). "L1-regularization path algorithm for generalized linear models." *Journal of the Royal Statistical Society, B*, **64**, 659–677.

Park MY, Hastie T (2007b). *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.

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/