A tutorial on fitting ordinal response models in high-dimensional datasets with the ordinalgmifs package

Kellie J. Archer*, Jiayi Houb, Qing Zhoua, Kyle Ferbera, John G. Laynea, Amanda Gentrya
* The Ohio State University, a Virginia Commonwealth University, b University of California San Diego

Abstract

In this tutorial we describe our ordinalgmifs R package, available from the Comprehensive R Archive Network, that can fit a variety of ordinal response models when the number of predictors (P) exceeds the sample size (n). We then illustrate the functions in the ordinalgmifs R package using a dataset where we were interested in predicting normal < pre-neoplastic < neoplastic states of liver disease using a subset of CpG sites from a high-throughout methylation assay (Archer et al. 2010).

Keywords: ordinal response, high-dimensional features, penalized models, R.

1. Introduction

Various algorithms can be used for obtaining solutions for the Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani 1996, 1997) and elastic net penalized models (Zou and Hastie 2005). In the linear regression setting, the Incremental Forward Stagewise (IFS) is a penalized solution that enforces monotonicity (Hastie et al. 2007). IFS can be generalized to problems involving other than squared error loss, and the adaption is called the generalized monotone incremental forward stagewise (GMIFS) method (Hastie et al. 2007). Herein we extended the GMIFS method (Hastie et al. 2007) to ordinal response setting and implemented various functions in our ordinalgmifs R package (Archer et al. 2014). The ordinalgmifs function can be used to fit traditional and penalized cumulative link, forward continuation ratio, and backward continuation ratio models using either a logit, probit, or complementary log-log link. It can also be used to fit adjacent category and stereotype logit models. A detailed description of the methodology is available in (Archer et al. 2014).

2. Implementation

The ordinalgmifs package was written in the R programming environment (R Core Team 2013). The ordinalgmifs function allows the user to specify a model formula, identify the matrix of covariates to be penalized in the model fitting algorithm using the x parameter, and additionally specify the model type (probability.model) and link function (link). The default is to fit a cumulative logit model though allowable probability models include "Cumulative", "ForwardCR", "BackwardCR", "AdjCategory", and "Stereotype" while al-
allowable links include "logit", "probit", "cloglog" for the first three and "loge" and "logit" for the last two, respectively. The defaults for updating the penalized coefficients are epsilon=0.001 and tol=1e-5. Our likelihood functions were written in R and tested by comparing our R output to output produced by the vglm R VGAM package for cumulative link, adjacent category, forward and backward continuation ratio models and to STATA’s slogit function and the rrvglm function in the R VGAM for the stereotype logit model using benchmark datasets for data where \( P < n \).

3. Examples

The ordinalgmifs package includes example datasets having an ordinal response. These data are a subset of subjects and CpG sites reported in the original paper where liver samples were assayed using the Illumina GoldenGate Methylation BeadArray Cancer Panel I (Archer et al. 2010). Technical replicate samples and matched cirrhotic samples from subjects with hepatocellular carcinoma (HCC) were removed to ensure all samples were independent. For the following examples, methylation levels for 45 CpG sites in liver tissue are provided for independent subjects whose liver was either Normal (N=20), cirrhotic but not having HCC (N=16, Cirrhosis non-HCC), and HCC (N=20, Tumor). These data are in two formats: a data.frame (hccframe) and as a BioConductor ExpressionSet (hccmethyl). We seek to predict group which indicates sample type (Normal < Cirrhosis non-HCC < Tumor) using the CpG site methylation values as predictor variables.

3.1. Model Fitting for Data Stored in a data.frame

To fit a model where all predictors are penalized the model formula is specified to fit an intercept only model and the predictors to be penalized are specified using the x parameter. When fitting a penalized model it is expected that more than one variable is included in the x parameter. The x parameter can either be a vector naming columns in the data.frame specified by the data parameter or x can be a the data.frame name with the columns to include (or exclude) indicated by their (negative) index. Prior to model fitting NA values should be imputed or removed from the data.frame. By default a cumulative logit model is fit when neither probability.model nor link are specified by the user. Because group is the first variable in the hccframe, we fit a model penalizing all CpG sites by specifying x=hccframe[,-1] which simply removes our ordinal outcome. Due to time constraints imposed by CRAN, we changed the default epsilon = 0.001 to reduce run time of this vignette.

```r
> library("ordinalgmifs")
> data(hccframe)
> cumulative.logit<-ordinalgmifs(group ~ 1, x = hccframe[,-1],
+     data = hccframe, epsilon=0.01)
```

Because the GMIFS procedure is incremental, the user may want to specify verbose=TRUE to print the step number in order to monitor the status of the model fitting procedure. Methods including coef, plot, predict, fitted, print, and summary can be applied to ordinalgmifs model objects. Because the returned list differs depending on whether a no penalty subset is included or a stereotype logit model is fit, the print function returns the object names of the fitted object.
By default, `coef`, `predict`, and `summary` extracts the relevant information from the step in the solution path that attained the minimum AIC.

```r
> summary(cumulative.logit)

Cumulative model using a logit link at step = 1521
logLik    = -1.529793
AIC       = 19467.06
BIC       = 21291.56

(Intercept):1                         (Intercept):2  CDKN2B_seq_50_S294_F
   -5.217902                             5.001486    -1.300000
DDIT3_P1313_R                          ERN1_P809_R   GML_E144_F
   -1.290000                             0.360000     1.920000
HDAC9_P137_R                           HLA.DPA1_P205_R  HOXB2_P488_R
    0.080000                            -0.360000    -0.080000
IL16_P226_F                            IL16_P93_R   IL8_P83_F
    1.820000                             0.340000     0.380000
MP0_E302_R                             MPD_P883_R   PADI4_P1158_R
    0.720000                             0.170000    -0.960000
SOX17_P287_R                           TJP2_P518_F   WRN_E57_F
   -1.940000                            -2.050000     0.540000
CRIP1_P874_R                           SLC22A3_P634_F  CCNA1_P216_F
    0.000000                             0.000000     0.000000
SEPT9_P374_F                           ITGA2_E120_F   ITGA6_P718_R
    0.000000                             0.000000     0.000000
HGF_P1293_R                            DLG3_E340_F   APP_E8_F
    0.000000                             0.000000     0.000000
SFTPB_P689_R                           PENK_P447_R   COMT_E401_F
    0.310000                             0.000000     0.590000
NOTCH1_E452_R                          EPHA8_P456_R   WT1_P853_F
    0.000000                             0.000000     0.000000
KLK10_P268_R                           PCDH1_P264_F   TDGF1_P428_R
    0.000000                             0.000000     0.000000
EFNB3_P442_R                           MMP19_P306_F   FGFR2_P460_R
    0.000000                             0.000000     0.000000
RAF1_P330_F                            BMP2_E435_F   GRB10_P496_R
    0.000000                             0.000000     0.000000
However, any step along the solution path can be extracted by specifying the step using the `model.select` parameter for these three functions. For example, the model attaining the minimum BIC can be extracted using `summary(cumulative.logit, model.select=which.min(cumulative.logit$BIC))`. Alternatively, the 150th step can be extracted using `summary(cumulative.logit, model.select=150)`. Note that the $\alpha_j$ thresholds are labelled as (Intercept):1,...,(Intercept):K-1.

The `plot` function plots the solution path of the model fit. The vertical axis can be changed using the `type` parameter with allowable selections being "trace" (default), "AIC", "BIC" or "logLik". Although there are default x-axis, y-axis, and titles provided for each plot, the user can modify these by supplying their own arguments to `xlab`, `ylab`, and `main`, respectively.

The `coef` function extracts the estimated parameters and returns them as a vector.

```r
> coef(cumulative.logit)

(Intercept):1  (Intercept):2  CDKN2B_seq_50_S294_F
   -5.217902       5.001486       -1.300000
DDIT3_P1313_R  ERN1_P809_R  GML_E144_F
   -1.290000       0.360000       1.920000
HDAC9_P137_R  HLA.DPA1_P205_R  HOXB2_P488_R
    0.080000       0.360000      -0.080000
IL16_P226_F  IL16_P93_F  IL8_P83_F
    1.820000       0.340000       0.380000
MP0_E302_R  MP0_P883_R  PADI4_P1158_R
    0.720000       0.170000      -0.960000
SOX17_P287_R  TJP2_P518_F  WRN_E57_F
   -1.940000      -2.050000       0.540000
CRIP1_P874_R  SLC22A3_P634_F  CCNA1_P216_F
   0.000000       0.000000       0.000000
SEPT9_P374_F  ITGA2_E120_F  ITGA6_P718_R
   0.000000       0.000000       0.000000
HGF_P1293_R  DLG3_E340_F  APP_E8_F
   0.000000       0.000000       0.000000
SFTPB_P689_R  PENK_P447_R  COMT_E401_F
   0.310000       0.000000       0.590000
NOTCH1_E452_R  EPHA8_P456_R  WT1_P853_F
   0.000000       0.000000       0.000000
KLK10_P268_R  PCDH1_P264_F  TGF1_P428_R
   0.000000       0.000000       0.000000
EFNB3_P442_R  MMP19_P306_F  FGFR2_P460_R
   0.000000       0.000000       0.000000
```
> plot(cumulative.logit)

![Cumulative model using a logit](image)

Figure 1: Coefficient estimates along the solution path for a fitted ordinalgmifs object using the hccframe data.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAF1_P330_F</td>
<td>0.000000</td>
</tr>
<tr>
<td>BMPR2_E435_F</td>
<td>0.000000</td>
</tr>
<tr>
<td>GRB10_P496_R</td>
<td>0.000000</td>
</tr>
<tr>
<td>CTS_H_P238_F</td>
<td>0.000000</td>
</tr>
<tr>
<td>SLC6A8_seq_28_S227_F</td>
<td>0.000000</td>
</tr>
<tr>
<td>PLXDC1_P236_F</td>
<td>0.000000</td>
</tr>
<tr>
<td>TFE3_P421_F</td>
<td>0.000000</td>
</tr>
<tr>
<td>TSG101_P139_R</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

The predict function (or equivalently, fitted) returns a list containing predicted, a matrix of the class probabilities from the fitted model, and class, the class having the maximum predicted probability from the fitted model. As with coef and summary the predict function by default extracts the model that attained the minimum AIC, but predictions for any step along the solution path can be obtained by specifying the step using the model.select parameter.
> phat <- predict(cumulative.logit)
> table(phat$class, hccframe$group)

<table>
<thead>
<tr>
<th></th>
<th>Normal</th>
<th>Cirrhosis</th>
<th>non-HCC</th>
<th>Tumor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cirrhosis</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>non-HCC</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

> head(phat$predicted)

[,1] [,2] [,3]
[1,] 4.124571e-08 0.0011300454 0.9988699
[2,] 3.752721e-06 0.0933262236 0.9066700
[3,] 2.010174e-09 0.0000551338 0.9999449
[4,] 1.193245e-06 0.0316920700 0.9683067
[5,] 2.335748e-06 0.0602093838 0.9397883
[6,] 1.351532e-06 0.0357458463 0.9642528

When there are small sample sizes in one or more groups K-fold cross-validation (CV) methods may not perform well as a means to estimate generalization error due to the random inclusion of samples into each of the folds. That is, multiple folds may include few if any subjects from the small classes. Therefore here we have demonstrated N-fold CV for this dataset. Note that we include the drop=FALSE argument to preserve the dimension format of the object when only one subject comprises the testset.

class<-character()
for (i in 1:dim(hccframe)[1]) {
  fit<-ordinalgmifs(group ~ 1, x = hccframe[-i,-1], data = hccframe[-i,])
  class[i]<-predict(fit, newx=hccframe[i,-1,drop=FALSE])$class
}
table(class, hccframe$group)

which yields a generalized misclassification rate of 10.7%.

The National Institutes of Health released notice NOT-OD-15-102 detailing the requirement for researchers to consider sex as a biological variable, which may lead the analyst to coerce sex into the multivariable model. There are a multitude of clinical scenarios where it is of primary interest to discover the additional predictive value of including molecular features beyond already known risk factors. Therefore, we extended our method to penalize some covariates (high-throughput genomic features) without penalizing others (such as demographic and/or clinical covariates). The following examples are merely to illustrate additional flexibility of the package (Gentry et al. 2015). Suppose that DDIT3_P1313_R is to be coerced into the model and only CDKN2B_seq_50_S294_F, ERN1_P809_R, GML_E144_F, and HDAC9_P137_R are to be penalized (the model includes only the 5 CpG sites). Any variable(s) to be coerced into the model should appear on the right-hand side of the model formula and represents the “unpenalized predictors”. The model can be fit using
cumulative.logit.2 <- ordinalgmifs(group ~ DDIT3_P1313_R,  
   x = c("CDKN2B_seq_50_S294_F", "ERN1_P809_R", "GML_E144_F",  
      "HDAC9_P137_R"), data = hccframe)  
summary(cumulative.logit.2)

Predictions can be obtained using

phat<-predict(cumulative.logit.2)
head(phat$class)
head(phat$predicted)

When predicting the outcome for a new set of observations, the predict function will accept arguments for newx (the penalized predictors), neww (the unpenalized predictors using model formula notation, and newdata (new data.frame that contains the unpenalized predictors). Suppose we want to leave observation 1 out, fit the model, then predict the class for observation 1 as a left out test set where we have coerced DDIT3_P1313_R in the model by including it as an unpenalized predictor. The following code would be used:

cumulative.logit.m1 <- ordinalgmifs(group ~ DDIT3_P1313_R,  
   x = c("CDKN2B_seq_50_S294_F", "ERN1_P809_R", "GML_E144_F",  
      "HDAC9_P137_R"), data = hccframe[-1,])
predict(cumulative.logit.m1, neww=~DDIT3_P1313_R, newdata=hccframe[1,],  
   newx=hccframe[1,c("CDKN2B_seq_50_S294_F", "ERN1_P809_R",  
      "GML_E144_F", "HDAC9_P137_R")])

Aside from a logit link, a probit or complementary log-log link can be used in conjunction with the cumulative link probability model. Here we include only the first five CpG sites to reduce computational time for this illustration. These three links are also available for probability.model="ForwardCR" and probability.model="BackwardCR". We previously demonstrated use of the forward continuation ratio model with a complementary log-log link for modeling a discrete survival outcome (Ferber and Archer 2015).

A stereotype logit model only uses a logit link while an adjacent category model only uses a log link. Misspecifying the link for either a stereotype logit or adjacent category yields a warning that is printed to the R console but only the correct link is used in the model fit. The following example illustrates specifying the data frame using data, the probability model using probability.model, and the link function using link.

adj.cat<-ordinalgmifs(group ~ 1, x = hccframe[, 2:6],  
   data = hccframe, probability.model = "AdjCategory", link = "loge")
summary(adj.cat)
phat.adj <- predict(adj.cat)
table(phat.adj$class, hccframe$group)

Acknowledgments

Research reported in this tutorial was supported by the National Library Of Medicine of the National Institutes of Health under Award Number R01LM011169. The content is solely
the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References


Affiliation:
Kellie J. Archer
Division of Biostatistics
College of Public Health
The Ohio State University
1841 Neil Ave.
240 Cunz Hall
Columbus, OH 43210
E-mail: archer.43@osu.edu
URL: https://cph.osu.edu/people/karcher
Jiayi Hou  
Clinical and Translational Research Institute  
University of California San Diego  
San Diego, CA  
E-mail: jhou@ucsd.edu

Qing Zhou  
Department of Biostatistics  
Virginia Commonwealth University  
Box 980032  
Richmond, VA 23298-0032  
E-mail: zhouq3@mymail.vcu.edu

Kyle Ferber  
Department of Biostatistics  
Virginia Commonwealth University  
Box 980032  
Richmond, VA 23298-0032  
E-mail: ferberkl@mymail.vcu.edu

John G. Layne  
Center for the Study of Biological Complexity  
Virginia Commonwealth University  
Box 842537  
Richmond, VA 23298  
E-mail: laynejg@vcu.edu

Amanda Gentry  
Department of Biostatistics  
Virginia Commonwealth University  
Box 980032  
Richmond, VA 23298-0032  
E-mail: gentryae@mymail.vcu.edu