

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

Kellie J. Archer^{*}, Jiayi Hou^b, Qing Zhou^a, Kyle Ferber^a, John G. Layne^a, Amanda Gentry^a

^{*} 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.¹

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)^{2,3} and elastic net penalized models.⁴ In the linear regression setting, the Incremental Forward Stagewise (IFS) is a penalized solution that enforces monotonicity.⁵ 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.⁵ Herein we extended the GMIFS method⁵ to ordinal response setting and implemented various functions in our **ordinalgmifs** R package.⁶ The `ordinal.gmifs` 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.⁶

2. Implementation

The **ordinalgmifs** package was written in the R programming environment.⁷ The `ordinal.gmifs` 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 allowable links include "logit", "probit", "cloglog" for the first three and "loge" and "logit" for the last two, respec-

tively. 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.¹ 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.

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

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.

```
> print(cumulative.logit)
```

```

[1] "beta"          "alpha"          "zeta"
[4] "x"             "y"             "w"
[7] "scale"         "logLik"         "AIC"
[10] "BIC"           "model.select"   "probability.model"
[13] "link"

```

By default `coef`, `predict`, and `summary` extracts the relevant information from the step in the solution path that attained the minimum AIC.

```
> summary(cumulative.logit)
```

Cumulative model using a logit link

```

at step    = 15062
logLik      = -1.568369
AIC         = 138315.1
BIC         = 151280.1

```

(Intercept):1	(Intercept):2	CDKN2B_seq_50_S294_F
-5.172655	4.950386	-1.287000
DDIT3_P1313_R	ERN1_P809_R	GML_E144_F
-1.283000	0.362000	1.906000
HDAC9_P137_R	HLA.DPA1_P205_R	HOXB2_P488_R
0.083000	0.354000	-0.076000
IL16_P226_F	IL16_P93_R	IL8_P83_F
1.805000	0.338000	0.381000
MPO_E302_R	MPO_P883_R	PADI4_P1158_R
0.715000	0.171000	-0.942000
SOX17_P287_R	TJP2_P518_F	WRN_E57_F
-1.925000	-2.023000	0.530000
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
0.300000	0.000000	0.581000
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	BMPR2_E435_F	GRB10_P496_R
0.000000	0.000000	0.000000
CTSH_P238_F	SLC6A8_seq_28_S227_F	PLXDC1_P236_F
0.000000	0.000000	0.000000

TFE3_P421_F	TSG101_P139_R
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 α_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.

```
> plot(cumulative.logit)
```

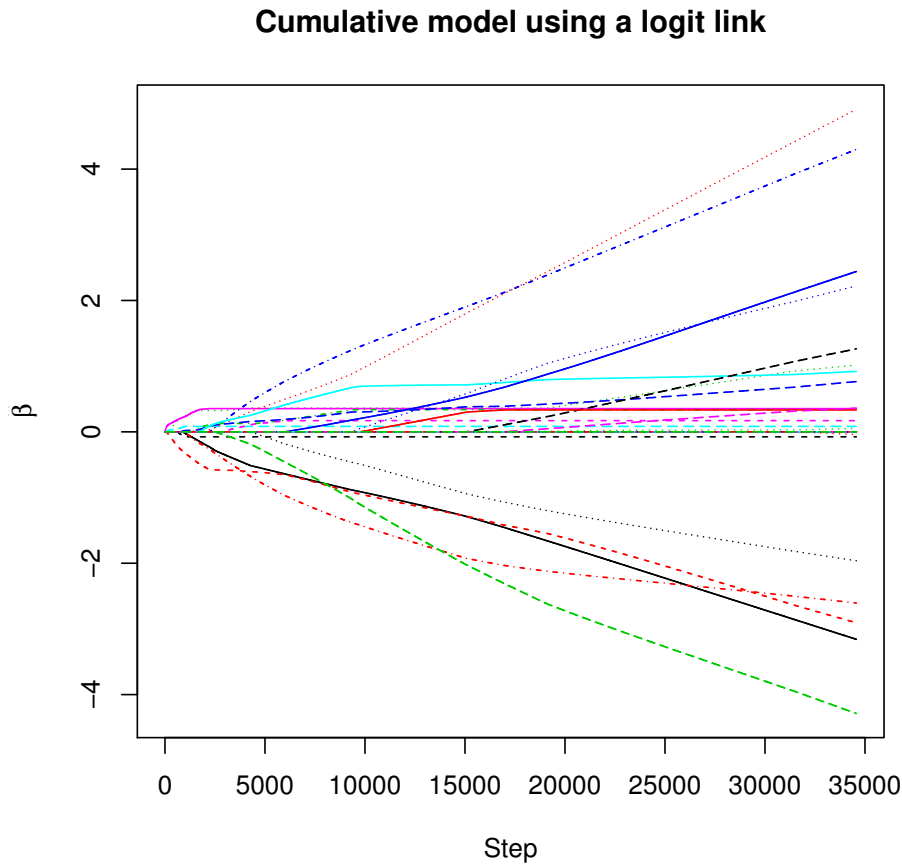


Figure 1: Coefficient estimates along the solution path for a fitted `ordinalgmifs` object using the `hccframe` data.

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)
```

	Normal	Cirrhosis	non-HCC	Tumor
Normal	20		0	0
Cirrhosis	0		16	0
Tumor	0		0	20

```
> head(phat$predicted)
```

	[,1]	[,2]	[,3]
[1,]	4.685414e-08	1.165749e-03	0.9988342
[2,]	4.206633e-06	9.484594e-02	0.9051499
[3,]	2.359269e-09	5.876458e-05	0.9999412
[4,]	1.357653e-06	3.271204e-02	0.9672866
[5,]	2.600891e-06	6.084460e-02	0.9391528
[6,]	1.542192e-06	3.699393e-02	0.9630045

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<-ordinal.gmifs(group ~ 1, x = hccframe[-i,-1],
+     data = hccframe[-i,])
+   class[i]<-predict(fit, newx=hccframe[i,-1,drop=FALSE])$class
+ }
> table(class, hccframe$group)
```

class	Normal	Cirrhosis	non-HCC	Tumor
1	20		1	0
2	0		13	3
3	0		2	17

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.⁸ 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 <- ordinal.gmifs(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)
```

Cumulative model using a logit link

```
at step    = 9326
logLik     = -12.15566
AIC        = 37338.31
BIC        = 40836.03
```

(Intercept):1	(Intercept):2	DDIT3_P1313_R
7.566608	17.555263	-21.268599
CDKN2B_seq_50_S294_F	ERN1_P809_R	GML_E144_F
-2.092000	2.253000	2.217000
HDAC9_P137_R		
2.764000		

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

```
> coef(cumulative.logit.2)
```

(Intercept):1	(Intercept):2	DDIT3_P1313_R
7.566608	17.555263	-21.268599
CDKN2B_seq_50_S294_F	ERN1_P809_R	GML_E144_F
-2.092000	2.253000	2.217000
HDAC9_P137_R		
2.764000		

Predictions can be obtained using

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

```
[1] Tumor Tumor Tumor Tumor Tumor Tumor
Levels: Normal < Cirrhosis non-HCC < Tumor
```

```
> head(phat$predicted)
```

```
      [,1]      [,2]      [,3]
[1,] 3.051490e-11 6.645219e-07 0.9999993
[2,] 1.158009e-06 2.459762e-02 0.9754012
[3,] 6.757248e-11 1.471522e-06 0.9999985
[4,] 8.735450e-06 1.598261e-01 0.8401652
[5,] 3.104283e-05 4.033356e-01 0.5966334
[6,] 2.234515e-07 4.842533e-03 0.9951572
```

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 <- ordinal.gmifs(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")]))
```

```
$predicted
      [,1]      [,2]      [,3]
[1,] 7.163164e-11 1.548215e-06 0.9999985
```

```
$class
[1] Tumor
Levels: Normal < Cirrhosis non-HCC < Tumor
```

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.⁹

A stereotype logit model only uses a logit link while an adjacent category model only uses a \log_e 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<-ordinal.gmifs(group ~ 1, x = hccframe[, 2:6],
+ data = hccframe, probability.model = "AdjCategory", link = "loge")
> summary(adj.cat)
```

AdjCategory model using a loge link

```
at step      = 9150
logLik       = -12.32746
AIC          = 36638.65
BIC          = 40070.75
```

(Intercept):1	(Intercept):2	CDKN2B_seq_50_S294_F
3.963392	-3.112291	1.878000
DDIT3_P1313_R	ERN1_P809_R	GML_E144_F
1.720000	-1.592000	-1.757000
HDAC9_P137_R		
-2.203000		

```
> phat.adj <- predict(adj.cat)
> table(phat.adj$class, hccframe$group)
```

	Normal	Cirrhosis	non-HCC	Tumor
Normal	20		1	0
Cirrhosis	0		13	3
Tumor	0		2	17

3.2. Model Fitting for Data Stored in an ExpressionSet

`hccmethyl` is a BioConductor ExpressionSet version of the data in `hccframe`. The **Biobase** BioConductor package is required to access the components of this object. Once R has been installed, open R by and install the *biocLite* script which will install a subset of the most frequently used Bioconductor packages. From the R prompt type,

```
> source("http://www.bioconductor.org/biocLite.R")
```

then

```
> biocLite()
```

Once installed, the **Biobase** package should be loaded. CpG site methylation data can be extracted using `exprs(hccmethyl)` and yields a matrix with 45 rows (CpG sites) and 56 columns (samples). Phenotypic data can be extracted using `pData(hccmethyl)`. Again to minimize processing time, only a subset of 5 CpG sites are included in this illustration.

```
> library("Biobase")
> data(hccmethyl)
> hccexprs.fit<-ordinal.gmifs(group ~ 1, x=t(exprs(hccmethyl)[1:5, ]),
+ data = pData(hccmethyl))
> coef(hccexprs.fit)
```



```

      (Intercept):1      (Intercept):2 CDKN2B_seq_50_S294_F
            -5.051716            3.865584            -2.297000
DDIT3_P1313_R      ERN1_P809_R      GML_E144_F
            -2.192000            1.828000            2.228000
      HDAC9_P137_R
            2.831000

> phat<-predict(hccexprs.fit)
> table(phat$class, pData(hccmethy1)$group)

```

	Normal	Cirrhosis	non-HCC	Tumor
Normal	20		1	0
Cirrhosis non-HCC	0		12	2
Tumor	0		3	18

The BioConductor package `GEOquery` can be used to download the full dataset, GSE18081.

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

- ¹ Archer KJ, Mas VR, Maluf DG, Fisher RA. High-throughput assessment of CpG site methylation for distinguishing between HCV-cirrhosis and HCV-associated hepatocellular carcinoma *Molecular Genetics and Genomics*. 2010;283:341–349.
- ² Tibshirani R. Regression shrinkage and selection via the Lasso *Journal of the Royal Statistical Society, B*. 1996;58:267–288.
- ³ Tibshirani R. The lasso method for variable selection in the Cox model *Statistics in Medicine*. 1997;16:385–395.
- ⁴ Zou H, Hastie T. Regularization and variable selection via the elastic net *Journal of the Royal Statistical Society B*. 2005;67:301–320.
- ⁵ Hastie T, Taylor J, Tibshirani R, Walther G. Forward stagewise regression and the monotone lasso *Electronic Journal of Statistics*. 2007;1:1–29.
- ⁶ Archer KJ, Hou J, Zhou Q, Ferber K, Layne JG, Gentry AE. **ordinalgmifs**: An R Package for Ordinal Regression in High-dimensional Data Settings *Cancer Informatics*. 2014;13:187–195.
- ⁷ R Core Team . *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing Vienna, Austria 2013.

⁸ Gentry AE, Jackson-Cook C, Lyon D, Archer KJ. Penalized Ordinal Regression Methods for Predicting Stage of Cancer in High-Dimensional Covariate Spaces *Cancer Informatics*. 2015;14(Suppl 2):201-208.

⁹ Ferber K, Archer KJ. Modeling discrete survival time using genomic feature data *Cancer Informatics*. 2015;14(Suppl 2):37-43.

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