PROC SCADGLIM User's Guide
Version 1.2 BETA

John J. Dziak, David R. Lemmon, Runze Li, and Liying Huang

September 4, 2014

Please send questions and comments to John Dziak at jjd264@psu.edu or Runze Li at rli@stat.psu.edu.

The authors thank Michael Yiyun Zhang, Stephanie Lanza and Joe Schafer for their valuable help.

Research and software development related to PROC SCADLS was supported by National Institute on Drug Abuse / National Institutes of Health grant P50 DA10075, and National Science Foundation (NSF) grants DMS-03048869, CCF 0430349 and DMS 0722351.

Contents

The SCADGLIM procedure is a SAS® implementation for the SCAD variable selection method, in the context of penalized maximum-likelihood generalized linear regression models. It was written for SAS® versions 9.1 or higher for Windows®.[1] Its goal is to select a parsimonious and well-fitting subset of a large number of potential predictor variables for a regression model with binary (logistic) or count (Poisson) responses, and automatically fit an adjusted generalized linear regression equation to this subset.

Section 2 explains the syntax for using PROC SCADGLIM, and section 3 describes the special case where some variables in the model are categorical rather than numerical. Appendix A provides detailed examples of using PROC SCADGLIM.

Version Note: The previous version of PROC SCADLS and PROC SCADGLIM, version 1.1 released in May 2010, was for 32-bit SAS installations only. The current version implements some minor bug fixes and is available for both 32-bit and 64-bit SAS installations.

Installation Note: The Methodology Center distributes an installation script which automatically puts the required files for PROC SCADLS and PROC SCADGLIM in the needed parts of your SAS folder so that the PROC's will work much like any built-in SAS PROC. However, for some people the installation script might have difficulty finding the folder (directory or path) on your computer which contains the SAS installation. In this case, you might have to provide the folder. That means you may have to type in the name of the path where your `sas.exe` executable file is found. It might be something like: `C:\Program Files\SASHome\SASFoundation\9.3`.

---

[1] SAS® software, Version 9.1 or higher of the SAS System for Windows. Copyright ©2006-2014 SAS Institute Inc. SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc., Cary, NC, USA. Windows® is a trademark of Microsoft Corporation.

1 SCAD-Penalized General Linear Regression Models

PROC SCADGLIM is an add-on procedure for SAS that performs SCAD-penalized maximum likelihood regression estimation for generalized linear models, allowing continuous, count or dichotomous outcomes. (For continuous outcomes, the companion procedure PROC SCADLS is recommended instead, since it is offers the faster ICM algorithm in such models).

PROC SCADGLIM is similar to PROC SCADLS in that it performs regression variable selection and estimation. However, SCADLS applies only to linear models, while SCADGLIM can handle linear models as well as two of the most common kinds of generalized linear models: binary regression with the logit link function and Poisson regression with the log link function. These three models assume that an individual's response Y_i depends on predictors X_1,...,X_p (actually, on some subset of the predictors, which SCAD attempts to identify) , and is independent of the responses of other individuals (or is at least conditionally independent given the predictors, if the predictors are treated as random). They differ in the form of the distribution which is assumed for Y:

Continuous (Normal Linear Regression) $Y_i = \mu_i + \epsilon_i$ where the errors ϵ_i are independent and normally distributed with mean 0 and variance σ^2, and where

$$\mu_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p. \tag{1}$$

This is similar to the model fit to normal data by the native SAS procedures PROC REG or PROC GLM, or to the native SAS procedure PROC GENMOD with the DIST=NORMAL option.

Count (Poisson Regression) Y_i has a Poisson distribution with mean μ_i, a positive number, where

$$\mu_i = \exp\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p\right) \tag{2}$$

This model uses a "log link" since log μ_i is modeled as a linear combination of the predictors. It is similar to the model fit by PROC GENMOD with the DIST=POISSON option.

Binary (Logistic Regression) Y_i is either 0 or 1. Its probability of being 1 is μ_i, a number between 0 and 1, where

$$\mu_i = \frac{\exp\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p\right)}{1 + \exp\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p\right)} \tag{3}$$

This model uses a "log link" since logit μ_i $(= \log\mu_i/(1 - \mu_i))$ is modeled as a linear combination of the predictors. This is similar to the model fit to binary data by the native SAS procedure PROC LOGISTIC, or by PROC GENMOD with the DIST=BINOMIAL option.

The practical difference between the models fit by SCADGLIM, and their equivalents in, say, PROC GENMOD, is that the SCAD procedure assumes that some of the β coefficients are zero or practically zero (at the population or "true value" level), and attempts to make the model more parsimonious by automatically identifying and setting these coefficients to zero. The classic model-fitting approach does not set any coefficients to zero, so that variable subset selection can only be done in a separate step. Thus SCAD acts a little like stepwise or best-subsets regression, in choosing a subset of regressors, and then estimates their coefficients in a specialized way. Mathematically, the ordinary techniques choose the β's to maximize the fitted log-likelihood function, but SCADGLIM chooses the β's to maximize a "penalized" log-likelihood which takes into account not only in-sample fit but also the degree of parsimony of the model.

2 Syntax for PROC SCADGLIM

The syntax for the SCADGLIM procedure is shown below. Bold-face text indicates required syntax; normal text indicates non-required syntax. text enclosed in < and > indicate places where names or special keywords may be included, and defaults are underlined. The syntax is not case-sensitive.

PROC SCADGLIM DATA = < dataset name > `< options > ;`
MODEL < variable name > = < list of variable names > `< options > ;`
DISTRIBUTION `< NORMAL, BINARY or POISSON >;`
`FORCEIN < list of variable names >;`
`CLASS < list of variable names >; < options >`
`SELECTION < GCV or BIC >;`
`DF < COUNT or PROJECTION >;`
`CRITERION < value (default is` 10^{-9}`) >;`
`MAXITER < value (default is 250) >;`
`MINLAMBDA < value (default based on data) >;`
`MAXLAMBDA < value (default based on data) >;`
`GRIDSIZE < value (default is 400) >;`
`A < value (default is 3.70) >;`

The abbreviation **DIST** can also be used in place of **DISTRIBUTION**.

The default values for MINLAMBDA and MAXLAMBDA are calculated based on the data.

The main differences between SCADLS 1.2 and SCADGLIM 1.2 syntax are that SCADGLIM offers the LINK statement while SCADLS does not, and SCADLS offers the ALGORITHM statement while SCADGLIM does not. SCADGLIM also has an additional MODEL option, YDESCENDING, which SCADLS does not offer; and SCADLS offers the NOINT option which SCADGLIM does not offer.

2.1 Options

The available procedure options for use in the first line are DATA, OUTBETAS, OUTERRS, OUTINFO, OUTPRED, DETAILS, ROBUST and NOPRINT. The only mandatory "option" argument is DATA, the name of the SAS dataset to be analyzed. The dataset must contain at least two variables (i.e., columns). There must also be as many complete cases (i.e., subjects, observations, rows) as the number of variables plus two, in order for the results to be meaningful. That is, the current PROC SCADGLIM and PROC SCADLS do not work for very-high-dimensional ($p > n$) variable selection. Also, as of the current version of SCADGLIM the number of cases must be greater than the number of variables.

The other procedure options, which may be omitted, are as follows:

- *OUTBETAS = < dataset name >.* Specifies the name of a dataset which will be created, which will contain the coefficient estimates for all regression parameters (including the intercept, unless *NOINT* has been specified).

- *OUTERRS* = < *dataset name* >. Specifies the name of a dataset which will be created, which will contain the standard error estimates for all regression parameters (including the intercept, unless *NOINT* has been specified).

- *OUTINFO* = < *dataset name* >. Specifies the name of a dataset which will be created, which will contain some information about the solution, such as the number of iterations required.

- *OUTPRED* = < *dataset name* >. Specifies the name of a dataset which will be created, which will contain fitted values $\hat{y}$ for all observations in the original dataset. As in SCADLS, you can include extra lines in the original dataset with values of interest for the predictors but a missing value (".") for the response, in order to extract predicted values for these values of interest. For binary models, the predicted values will be numbers between 0 and 1 expressing, for each subject, estimated probability of observing Y=1 for a subject with the same values on the covariates.

- *ROBUST*. Causes "sandwich" standard errors, which may be more robust to heteroskedasticity, to be calculated in place of the usual model-based standard errors. This is only applicable when using a normal model (not binary or Poisson).

- *DETAILS*. Causes extra details about the fitting procedure to be displayed on standard output (i.e., the screen in Windows®, or the list file if one has been set). If the *DETAILS* option is not requested, only the most important basic information is shown.

- *NOPRINT*. Specifies that no information at all be displayed. This would mainly be of use in simulations or other automated analyses where the results of interest will be read from the output datasets.

Two options are available in the MODEL statement:

- *NOINT*. Specifies that no intercept parameter be included in the model (i.e., that the intercept be fixed at zero). By default (if this keyword is left out), there is an intercept. NOINT can only be specified if a normal (not binary or count) model is being used.

- *YDESCENDING*. This option can only be used if a binary regression with character-valued response variable is being performed. For these models, the response which comes last in alphabetical order is considered the success by default (the event being modeled, i.e., $Y = 1$). YDESCENDING reverses this convention and models the first event in alphabetical order as the success. For example, if the response variable has possible values "Yes" and "No," the fitted values would normally tell the estimated probability of the "Yes" response, but YDESCENDING would specify that it should be the "No" response.

There is also one CLASS statement option:

- *DESCENDING*. Specifies that the names of levels of categorical predictor variables be sorted in reverse alphabetical order, with the comparison group being the last in line (the first in alphabetical order). Otherwise, the SAS default is followed, and the levels are presented in alphabetical order, with the comparison group being the last in line (the last in alphabetical order).

The keywords in the rest of the syntax are as follows. The first two are required:

- **MODEL**. Specifies the full model, of which SCAD will be used to choose a subset. As in many SAS PROCedures such as PROC REG, it follows the form RESPONSE = PREDICTORS. It is very similar to the MODEL statement in SCADLS, as well as in various native SAS procedures for regression.
- **DISTRIBUTION**. Required. Specifies which kind of generalized linear model will be fit:

  NORMAL (synonyms: GAUSSIAN, IDENTITY) The ordinary linear model as in (1)

  POISSON (synonyms: COUNT, LOG) The log-linear Poisson model as in (2)

  BINARY (synonyms: LOGISTIC, LOGIT) The logistic binary model as in (3)

  Unlike in PROC GENMOD, in SCADGLIM one does not choose the response distribution and link function separately, since the canonical link function is assumed. That is, SCADGLIM assumes a linear model if the response is specified as normal, a log-linear model if the response is specified as Poisson, or a logistic model if the response is specified as binary.

The following statements are optional. Of them, FORCEIN and SELECTION are the most important to understand. The others are set at sensible defaults by the procedure and most users should not have to worry about them.

- *FORCEIN*. Optional. Names certain predictors which should always be included in the final model, i.e., are "forced in" regardless of the size of their coefficient estimate. They will not be penalized or deleted. Every variable listed under FORCEIN must also be listed as a predictor in the MODEL statement. You might want to use the FORCEIN keyword if some of your predictors are especially important for theoretical and practical reasons, or if submodels lacking a certain predictor of particular interest would be considered uninterpretable even if they did well statistically. For example, one of the predictor variables might be a pretest, or might be one of the main substantive focuses of the study. Note that if FORCEIN is not specified, no variables are forced in. Thus, all variables will be eligible for possible deletion if their coefficient estimates are judged non-significant by the algorithm. In the opposite extreme, if all variables are forced in, then the result is the ordinary maximum likelihood estimate for the full model. Note that there is no need for a "force out" option because variables which are known *a priori* to be useless may simply be left out of the MODEL statement.

- *CLASS.* Optional. Names certain predictors which should be treated as categorical, i.e., automatically recoded as dummy codes. The predictors may be text or numerical. However, since each observed level of the class variable, except for one baseline level, gets its own dummy-coded variable, a continuously measured numeric variable should never be specified as CLASS. The CLASS statement here works somewhat similarly to those found in PROC GLM, PROC GENMOD, PROC LOGISTIC, and PROC MIXED. This statement is the same as the CLASS statement in the PROC SCADLS manual and is described further there.

- *SELECTION.* Optional. If specified, it must be either AIC, or BIC. The default is BIC, as in SCADLS. This chooses whether the model should be more lightly penalized (AIC) or heavily penalized (BIC). The GCV option, which is rather similar to AIC, is also allowed in the NORMAL case for comparability with PROC SCADLS. **Note:** The AIC and BIC criteria are based on the negative two log-likelihoods, penalized according to the number of parameters. In the case of the Poisson distribution, the full -2 log likelihood is not calculated; specifically, the term $\sum \log(y!)$ is omitted to streamline computations because it depends only on the y values and does not depend on the model fit. This does not matter for model selection purposes but is useful to know if one is comparing BIC values between this PROC and other packages.

- *DF.* Optional. If specified, it should be either COUNT or PROJ. This chooses the way the size of the model should be measured while implementing the GCV and BIC criteria. The default option, COUNT, specifies that the model size should be treated simply as a count of nonzero coefficients. The other option, PROJECTION or PROJ for short (either will work), specifies that the model size should be measured in a way which takes shrinkage into account, by calculating the trace of the approximate linear projection matrix (i.e., a smoothing or "hat" matrix) resulting from the penalized estimation. The PROJ option is more compatible with Fan and Li (2001). However, the COUNT method is also reasonable (see Zou et al., 2004) and is much faster computationally, so it is the default. For a given candidate model, the PROJ option counts the selected model as being somewhat smaller than the COUNT option would; as a result, the COUNT option leads on average to the selection of a slightly smaller final model.

- *ALGORITHM.* Optional. If specified, it should be either LQA or ICM. This chooses which algorithm will be used to find the coefficient estimates. The LQA option is in keeping with Fan and Li (2001), but the ICM option is computationally faster. They should give essentially the same results, so the ICM option is the default.

- *CRITERION.* Optional. This defines the convergence criterion (i.e., tolerance). For a given value of λ, the LQA or ICM algorithm continues to iterate until either the maximum absolute difference between the coefficient estimates from the current and previous iterations is less than CRITERION, or the number of iterations is equal to MAXITER. The default is 10^{-9}, which is written as either .000000001 or 10E-9 in SAS. It can be made lower (for more precision) or higher (for faster speed). We recommend that CRITERION be no greater than 1E-7.

- *MAXITER.* Optional. For each value of λ in the grid of candidate lambdas, no more than MAXITER iterations will be performed, in order to save time. If the estimate has not converged yet after MAXITER iterations, it is left as it was on the (MAXITER)th iteration. Once the optimal lambda value has been chosen, the final model will be computed using up to 10×MAXITER iterations. If even this is not enough for the final model to fully converge, the answer will be reported anyway but a warning will be shown. We recommend that MAXITER be no less than 200 for PROC SCADGLIM.

- *MINLAMBDA.* Optional. This will be the least value of λ considered in the grid of candidate λ values. By default it will be automatically calculated as $\sigma/(20\sqrt{n})$, a very small amount of shrinkage, where σ is a rough estimate of the standard deviation of the response variable. You might want to reset it to a higher value to force a smaller model to be chosen. Alternatively, you might want to set it to zero, to allow the ordinary least-squares (full-model) estimate to be potentially chosen if it is the one with the best GCV or BIC.

- *MAXLAMBDA.* Optional. The counterpart to MINLAMBDA, this is the highest candidate value considered in the lambda grid. The default is $5\sigma\sqrt{\log(n)/n}$, a fairly large amount of shrinkage.

- *GRIDSIZE.* Optional. This is the number of candidate values considered in the lambda grid.

- *A.* Optional. This is a parameter that controls the shape of the SCAD penalty function. a must be at least 2 . The default value is 3.7 as in Fan and Li (2001). The effects of different a parameters in non-normal models have not yet been well studied.

Section 3 describes how to use the CLASS option to handle non-numerical predictor variables, and how to use non-numerical response variables in a BINARY model. Appendix A on page 12 provides detailed examples of using PROC SCADGLIM.

3 Categorical Predictors and Responses in SCADGLIM

Typically, both the predictor and response variables are numerical. However, categorical variables can be used as predictors using the CLASS statement, and in the case of a binary model, a categorical variable can also be used as the response.

The CLASS statement for categorical predictors works just as in PROC SCADLS and is described in the SCADLS users' manual.

Categorical responses, on the other hand, are only available with SCADGLIM and the BINARY model. Responses for the BINARY model can be represented either as dummy codes (0's and 1's, with 1 representing that the event of interest occurred for that subject and 0 representing that it did not) or as text (e.g., "Yes"/"No", or "Died"/"Survived"). In the case of text responses, SCADGLIM must determine which one represents the event of interest. It will ordinarily choose the last in alphabetical order (e.g., "Yes", "Survived") as the $Y = 1$ response but if the YDESCENDING option is specified then it will choose the first. If there are more than two values given (e.g., "Died"/"Injured"/"Survived") then all but the alphabetically last (or first, if YDESCENDING) value are lumped together as representing $Y = 0$.

References

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360, 2001.

H. Zou, T. Hastie, and R. Tibshirani. On the "degrees of freedom" of the Lasso. Technical report, Stanford University, 2004.

A Appendix: Example

In the following example, SCADGLIM is used to fit a logistic regression model. The data is artificial but was simulated to mimic some characteristics of a set of eighth graders from the public subset of the Add Health dataset on adolescent health behaviors, psychology and development in the United States.

The variables in the example dataset are:

male01	Gender	0=female, 1=male
eat_breakfast01	Eats breakfast regularly	0=no, 1=yes
intelligent1to6	Self-rated intelligence	1=low, 6=high
lowincome01	Family has low income	0=no, 1=yes
bills01	Family has problems paying bills	0=no, 1=yes
single01	Single parent	0=no, 1=yes
life_alc01	Ever used alcohol	0=no, 1=yes
yr_binge01	Binge drank in past year	0=no, 1=yes
life_cig01	Ever smoked cigarette	0=no, 1=yes
peer_alc01	One or more friends use alcohol	0=no, 1=yes
peer_cig01	One or more friends smoke cigarettes	0=no, 1=yes
peer_cig01	One or more friends use marijuana	0=no, 1=yes
insomnia01	Recently problems with insomnia	0=no, 1=yes
crying01	Recently problems with crying	0=no, 1=yes
fearful01	Recently problems with feeling fearful	0=no, 1=yes
sad0to19	Number of depression or sadness symptoms endorsed out of 19	0 to 19
intelligent1to6	Self-perceived intelligence	1 to 6 (3="average")
life_mar01	Ever used marijuana	0=no, 1=yes

Suppose we wanted to find a model that predicted lifetime use of marijuana from the other variables available. Suppose we want to force grade level and peer marijuana use into the model because we don't think that a model without those variables would make sense.

We could say

```
PROC SCADGLIM DATA=addhealthsim OUTBETAS=BETA1 OUTERRS=ERRS1 OUTPRED=FITS1;
     MODEL life_mar01 = male01 lowincome01 bills01
          single01 life_alc01 yr_binge01 life_cig01
          peer_cig01 peer_alc01 peer_mar01 eat_breakfast01
          insomnia01 crying01 fearful01 sad0to19 intelligent1to6 ;
     FORCEIN male01 peer_mar01 ;
     DISTRIBUTION BINARY;
RUN;
```

The results are

```
Number of observations:              2000
Number of predictors:                   16
Predictors forced in:                                    male01 peer_mar01
Model type:  Logistic (generalized linear model, binary outcome, logit link)

Selected lambda:                       0.0340
BIC:                                1433.1503

Converged in 15 iterations.

Final Estimates:

  Variable      Beta   Std. Errs.
-----------------------------------
Intercept :  -2.708179  0.383444
male01    :  -0.361896  0.140107
life_alc01:   0.982607  0.182844
yr_binge01:   0.534305  0.161926
life_cig01:   0.528153  0.173032
peer_cig01:   0.761980  0.150165
peer_alc01:   0.842113  0.166118
peer_mar01:   0.015923  0.023881
sad0to19  :   0.153151  0.042530
intelligen:  -0.262232  0.084853
```

Now suppose we select just the marijuana users and model their use as a Poisson count variable (although this is not strictly correct, since the subsample is defined in a way that makes a "0" count impossible).

```
PROC SCADGLIM DATA=users OUTBETAS=BETA1 OUTERRS=ERRS1 OUTPRED=FITS1;
     MODEL mar_uses_life = male01 lowincome01 bills01
          single01 life_alc01 yr_binge01 life_cig01
          peer_cig01 peer_alc01   eat_breakfast01
          insomnia01 crying01 fearful01 sad0to19 intelligent1to6 ;
     FORCEIN male01   ;
     DISTRIBUTION POISSON;
RUN;
```

The results are:

```
        PROC SCADGLIM -- Data and Model Summary and Fit Statistics

Number of observations:               401
Number of predictors:                  15
Predictor forced in:                                     male01
Model type:  Poisson (generalized linear model, count outcome, log link)
```

```
Selected lambda:                          0.3281
BIC:                                  -9106.4005

Converged in 203 iterations.

Final Estimates:

  Variable      Beta   Std. Errs.
-----------------------------------
Intercept :   1.075641  0.061733
male01         :   0.058516  0.034984
bills01        :   0.060515  0.024664
life_alc01     :   0.933460  0.060433
yr_binge01     :   0.312066  0.031061
sad0to19       :   0.042855  0.007743
```