PROC SCADLS 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 SCADLS procedure is a SAS® implementation for the SCAD-penalized variable selection method, in the context of linear regression models. It was written for SAS®, Version 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 linear regression, and automatically fit an adjusted regression equation to this subset.

SCAD-penalized regression, as described by Fan and Li (2001), is a way of constructing a regression model by doing predictor variable selection and coefficient estimation together. This is done by optimizing a penalized least squares criterion that expresses a balance between good fit and parsimony. The primary outcome is a list of regression coefficients which have been modified somewhat from the ordinary least-squares regression estimates. In particular, some of the new coefficients, whose least-squares estimates were near zero, are set to exactly zero in order to simplify the model. In other words, variable subset selection is performed automatically. This is somewhat similar to a thresholding approach in which coefficients judged insignificant are deleted from the model. However, SCAD applies a more nuanced approach; in particular, coefficients which were barely large enough not to be deleted are still shrunken somewhat (biased towards zero) in an attempt to reduce unnecessary sampling variance. Large coefficients are left close to their least-squares value under the selected model.

Key features of PROC SCADLS include the following:

- The penalty tuning parameter λ can be selected automatically using an adapted version of either the Generalized Cross-Validation (GCV) criterion or the the Schwarz Bayesian Information Criterion (BIC), depending on the user's choice. λ controls the number of variables in the selected subset, as well as the size of the shrinkage adjustment to their estimated coefficients (see Fan and Li, 2001).

- Variables of special theoretical or practical importance may be forced into the final model.

- Standard error estimates are calculated using a sandwich formula as in Fan and Li (2001).

- Categorical predictors can be included via a CLASS statement.

The previous version of PROC SCADLS, 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.

The SCAD estimation procedure differs from older methods (e.g., stepwise selection and best-subsets) in that some of the retained coefficients may be shrunken (biased towards zero). This could be thought of as adjusting for uncertainty in whether or not to exclude the variable. This is an attempt to reduce error variance and improve upon the

[1] SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Windows®is a trademark of Microsoft Corporation.

discontinuous and somewhat unstable nature of the older methods (see Breiman, 1995, 1996; Fan and Li, 2001).

Section 2 explains the syntax for using PROC SCADLS, and section 3 describes the special case where some predictor variables are categorical rather than numerical. Section 4 describes special considerations and situations. Appendix A on page 19 describes the theoretical background, properties, and implementation of SCAD in more detail; it may be skipped by a user who wishes to begin using PROC SCADLS quickly. Appendix B on page 30 provides detailed examples of using PROC SCADLS.

1 SCAD-Penalized Linear Regression

As mentioned above, PROC SCADLS is a new procedure for SAS that performs SCAD-penalized least squares regression estimation, particularly for linear models with roughly normally distributed outcomes. For categorical or count outcomes, one can use PROC SCADGLIM instead.

Like the classic multiple regression methods (such as those in PROC REG or PROC GLM in SAS), PROC SCADLS is intended to estimate the parameters of a linear model to predict the value of a response variable Y from predictor variables X_1,...,X_p. For example, Y could be a measure of a person's mood, income, or health, and X_1,...,X_p are various other variables which may be useful in predicting Y. Also like classical regression modeling, it is assumed that the value of Y is related to these predictors linearly: for a given case i, $Y_i = \mu_i + \epsilon_i$ where the ϵ_i have mean 0 and some constant variance σ^2, and

$$\mu_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p.$$

However, in the SCAD approach it is additionally assumed that some of the β's are equal to 0 (or at least so close to zero that they can be practically treated as such) while the others are not, and that it is desired to find out which β's are in each of these two categories. Thus, the goal is to select and estimate the nonzero coefficients while leaving the others at zero. To do this, the SCAD approach is to find the values of β_1,...,β_p which optimize a special function taking into account both how well the estimated model fits the in-sample data (in terms of the squared errors, for PROC SCADLS, or the likelihood, for PROC SCADGLIM), and the number and size of the nonzero coefficients.

Thus, SCAD is a variable selection method, similar in purpose to stepwise or best-subsets variable selection, since the predictors with zero coefficients are considered to be deleted from the model. However, SCAD is a somewhat different approach in that it treats selection and estimation together as a single optimization problem.

2 Syntax for PROC SCADLS

The syntax for the SCADLS procedure is shown below. Bold-face text indicates required syntax. Text enclosed in < and > indicate places where names or special keywords may be included. Default values are underlined. As usual in SAS, the syntax is not case-sensitive.

```
PROC SCADLS DATA = < dataset name >  < options > ;
   MODEL < variable name > = < list of variable names >  < options > ;
   FORCEIN < list of variable names >;
   CLASS < list of variable names >; < options >
   SELECTION < GCV or BIC >;
   DF < COUNT or PROJECTION >;
   ALGORITHM < ICM or LQA >;
   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 default values for MINLAMBDA and MAXLAMBDA are calculated based on the data.

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, which works just as in other procedures such as PROC PRINT, MEANS, REG, etc. The dataset must contain at least two variables (i.e., columns) and at least two complete cases (i.e., subjects, observations, rows) in order for the results to be meaningful. Also, as of the current version of SCADLS 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 predicted values $\hat{y}$ for all observations in the original dataset. You can then compute and analyze residuals by subtracting these from the observed values (i.e., $y - \hat{y}$). As with SAS PROC REG, 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. Predicted values are calculated as if the SCAD estimator $\hat{\boldsymbol{\beta}}$ were an ordinary regression estimate, i.e., $\hat{y}_i = \mathbf{x}_i^T \hat{\boldsymbol{\beta}}$.

- *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.

One option is 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.

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 is 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. RESPONSE is the name of the variable in the working dataset which will be treated as the response y (also known as the dependent variable or outcome). Only one variable may be specified as the RESPONSE at a time. PREDICTORS tells the names of the variables in the working dataset which will be treated as the predictors $x_1, ..., x_d$ (also known as the independent variables or covariates). At least one variable must be available as a PREDICTOR. Theoretically there is no upper limit on the number of predictors. (However, currently the SCADLS procedure only works when the number of predictors p is less than the number of observations n, since $x_1, ..., x_d$ are considered as predictors in the full model. The $p > n$ case, as in bioinformatics research, requires a different estimation approach which has not yet been implemented here (see Zou and Li, 2007).)

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 least-squares estimate for the full model. Note that there is no need for a "force out" option because variables which are known not to be of interest 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, it is quite inadvisable to use a continuously measured numeric variable. The CLASS statement here works somewhat similarly to those found in PROC GLM, PROC GENMOD, PROC LOGISTIC, and PROC MIXED. This is described further in Section 3.

- *SELECTION.* Optional. If specified, it must be either GCV or BIC. This chooses whether the model should be more lightly penalized (GCV) or heavily penalized (BIC). Fan and Li (2001) used GCV, but the default here is BIC because models selected by GCV are not as sparse (see Wang et al., 2007).

- *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 100 for ICM or 200 for LQA.

- *MINLAMBDA.* Optional. This will be the least value of λ considered in the grid of candidate λ values. By default it will be calculated as $\sigma/(20\sqrt{n})$, where σ is a rough estimate of the standard deviation of the response variable. This is a very small amount of shrinkage. 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 $5\sigma\sqrt{\log(n)/n}$.

- *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; as $a \to \infty$, SCAD acts more and more like LASSO. The default value is 3.7 as recommended in Fan and Li (2001).

To show how to use this syntax in practice, Section 3 on page 11 describes how to use the CLASS option to handle non-numerical predictor variables. Section 4 on page

14 discusses other special challenges in particular situations. Appendix B on page 30 presents examples using real and simulated datasets.

3 Categorical Predictors in PROC SCADLS: The CLASS Statement

As is well known, multiple linear regression methods, including SCAD-penalized regression, require that the predictor variables be numerical, but in many situations we also want to use categorical predictors. These predictors may be nominal (e.g., gender, ethnic group, religion, ice cream flavor) or ordinal (e.g., rating of conditions as poor/fair/good). Thus, we need a way to translate a categorical variable into one or more numerical ones. Simply assigning a number to each level is obviously inadequate. For instance, if we code Chocolate as 1, Strawberry as 2 and Vanilla as 3, and treat this as a numerical variable, then we seem to be asserting that strawberry is halfway between chocolate and vanilla on some numerical dimension of interest, which is arbitrary and incorrect.

The only time a categorical variable can be directly turned into a number is when it has only two possible levels. For example, if we code male as 0 and female as 1, we have reexpressed gender as a "femaleness" score which goes from low to high. This seems somewhat arbitrary, since we could have used -1 and 1, or 1 and 2, instead of 0 and 1; or we could have used 1 and 0 to get a "maleness" score. However, although each of these options changes the regression coefficients in a predictable way (e.g., the last one causes the sign to be reversed), none of them changes the assumptions, predicted values, or meaning of the model. In particular, because each coding has only two levels, we do not have to worry about which levels are between which other levels. The 0/1 coding scheme, called "indicator" or "dummy coding," is the most traditional in regression contexts (unlike ANOVA-like analyses of designed experiments in which -1/1 is preferred).

The usual way, then, to handle variables with $k \geq 2$ categories is to choose one level as a baseline or comparison group, and then express each other group as a dummy code. When we reexpressed gender as femaleness, we made "male" the baseline and we only needed one dummy variable to account for the one non-baseline group. If we had made female the baseline, the dummy coding would be reversed, but there would still only be one dummy code. In the ice cream example, suppose we make vanilla the baseline and create two new variables: chocolate-ness (1 for chocolate and 0 otherwise) and strawberry-ness (1 for strawberry and 0 otherwise). We do not need a vanilla-ness variable because the baseline is expressed by leaving all of the dummy codes at zero. In general, we need $k-1$ dummy codes to express k categories. Once this recoding is done, we can ignore the original categorical variable and use the $k-1$ dummy codes instead.

There are three complications here. The first is the necessity of *choosing a baseline*. In some cases the choice may be clear; in other cases it may be arbitrary. If you have a prior preference of which category should be the baseline, you may wish to do the dummy coding yourself. As an amusing toy example, suppose we are predicting mathematics achievement from verbal IQ, nonverbal IQ, and flavor preference.

```
DATA iceCream;
  INPUT flavor$10.  verbalIQ nonverbalIQ mathAchievement;
  DATALINES;

Vanilla    116 121 116
Chocolate  132 119 112
```

```
Vanilla    115 112 121
Strawberry 102 115 165
Vanilla    130 107 118
Chocolate   98 85 83
Chocolate  119 128 122
Vanilla    122 117 100
Strawberry 109 92 135
Vanilla    103 84 88
Strawberry 118 114 98
Vanilla    133 128 109
;
RUN;
```

We cannot enter the first variable directly into PROC SCADLS (or PROC REG, etc.) because it is a text field, not a number. However, we can dummy-code it:

```
DATA iceCream; SET iceCream;
likesChocolate = 1*( flavor= "Chocolate" );
likesStrawberry = 1*( flavor= "Strawberry" );
RUN;
```

In this case we are supposing that the baseline is vanilla so we do not need a dummy code for vanilla.

A second complication arises specifically with automated variable selection. Ordinarily there is no protection against some of the dummy codes of a predictor being deleted. (Cohen, 1991, discussed this problem in the context of stepwise selection.) In ordinary regression on the full model, the meaning of the model does not depend on the choice of baseline. However, if individual dummy-coded levels can be deleted, then the choice of baseline matters somewhat more (since level A might be significantly different from level B but not from level C). Choosing a very small baseline category may make estimates unstable and can lead to a poor choice of model.

Some SAS procedures include an option to create dummy codes automatically for the user's convenience, especially in situations in which the choice of baseline level is not of great interest. This is the CLASS statement in PROCs GENMOD, GLM, LOGISTIC and MIXED (such a statement is not found in the original PROC REG). There is also a similar CLASS statement in PROC SCADLS. This statement automatically generates dummy codes for categorical predictors. Similar to the SAS default, the dummy codes are generated in alphabetical order, and the baseline level is chosen to be the last in alphabetical order. (The DESCENDING option allows the baseline level to be the first in alphabetical order instead.)

However, we recommend writing one's own recoding statements for especially important predictors, and for predictors involved in interactions, because this requires the user to think about what the baseline should be. As mentioned earlier, the baseline is more important for SCADLS than for PROC's which do not delete predictors. There might be a theoretical reason to choose a baseline: for instance, we might want to use vanilla as

the baseline since it is the mildest flavor, or maybe the cheapest for a certain company to produce, or the most popular in the general population, etc. We do suggest that the baseline category be chosen to be one of relatively high frequency in the data, either the most common or one of the most common. Since the coefficients for the non-baseline categories are defined by contrast with the baseline, a baseline category with only a few observations might lead to a unstable model with high standard errors, and sometimes too many deletions. Similarly, in handling categorical predictors which are thought to interact, it is probably best to recode one's own predictor variables rather than let this be done automatically, and to include the relevant products of these recoded variables. In a model with interactions, an effect codes scheme such as -1/+1 may be preferable for reducing collinearity, in the same way that it may be helpful to center quantitative variables; see West et al. (1996) or Myers and Well (2003) for more information.

4 Tips for Special Situations

Several special challenges which sometimes arise in variable selection are discussed below. Some such issues are handled well by PROC SCADLS, while others may require additional work or different software.

4.1 Categorical Outcome Variables

Section 3 discussed how to deal with categorical predictor variables when the outcome is numerical. In other situations, the outcome variable (not just the predictors) is categorical. For instance, we may be predicting whether patients survive or die, given various characteristics of their condition. SCADLS cannot handle this kind of data, but its companion product SCADGLIM can do so.

4.2 Missing Data

PROC SCADLS deals with missing data as follows: For purposes of estimating the coefficients and standard errors, cases with any missing values in the response or any of the predictor variables are ignored (this is called "listwise deletion"). When calculating the predicted values for the OUTBETAS option, cases with missing values in the response, but no missing values in the predictor, are given a predicted value just like other cases. Those with missing values in the predictors are also given a missing value in the response. This is the same as the behavior of existing SAS regression procedures such as PROC REG. Of course, listwise deletion is not the best approach to missing data in general — it is often better to use a multiple imputation scheme when fitting a given model — but the best way to do this depends on the investigator's circumstances and assumptions, and the question of how to handle missing data in the context of model selection uncertainty requires more research. For more information see Collins et al. (2001), Little and Rubin (2002), and Schafer and Graham (2002).

4.3 Skew and Outliers

It is well-known that least-squares regression, as well as variable selection and testing techniques based upon it (including SCAD), are potentially vulnerable to model misspecification and to outliers. Robust methods are possible (see Fan and Li, 2001, for an example with SCAD in robust regression with L_1 predictive loss) but PROC SCADLS currently only supports SCAD-penalized least squares, which is much more straightforward computationally. Regardless of the method being used, it is always important to check the chosen model for notable violations of assumptions, especially by graphically examining the residuals. It is easy to calculate the residuals by subtracting the predicted values (given by OUTPRED) from the observed values. Another possibly helpful alternative is the ROBUST option in SCADGLIM.

4.4 Interactions and the Hierarchical Principle

Often, the full model (1) will include possible interactions, represented as products of observed predictors or dummy codes. Many investigators feel that if interactions are included in a model, the associated main effects must also be included. That is, suppose our predictors are x_1, x_2, and $x_3 \equiv x_1 x_2$. Then many people would say that a model which included x_3 will not be not very interpretable unless and x_1 and x_2 are also included. This becomes an issue when using SCAD, or other numerical variable selection methods, because it is quite possible for, say, x_2 to be deleted while keeping x_1 and x_3. One way to get around this is to perform selection in more than one step. For example, one might exclude all interactions at first and consider only the observed predictors. Predictors judged significant under this first wave would then be forced into a new model which also included all of their possible interactions, and SCAD-penalized regression would be applied again.

References

H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csaki, editors, *Second International Symposium on Information Theory*, pages 267–281. Akademai Kiado, Budapest, 1973.

A. Antoniadis and J. Fan. Regularization of wavelet approximations. *Journal of the American Statistical Association*, 96:939–955, 2001.

L. Breiman. Better subset regression using the nonnegative garotte. *Technometrics*, 37: 373–384, 1995.

L. Breiman. Heuristics of instability and stabilization in model selection. *Annals of Statistics*, 24:2350–2383, 1996.

K. P. Burnham and D. R. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer-Verlag, New York, 2nd edition, 2002.

K. P. Burnham and D. R. Anderson. Multimodel inference: Understanding AIC and BIC in model selection. *Sociological methods and research*, 33:261–304, 2004.

A. Cohen. Dummy variables in stepwise regression. *The American Statistician*, 45:226–228, 1991.

L. M. Collins, J. L. Schafer, and C. K. Kam. A comparison of inclusive and restrictive strategies in modern missing-dataprocedures. *Psychological Methods*, 6:330–351, 2001.

P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerical Mathematics*, 31:377–403, 1979.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

J. Fan. Comments on 'Wavelets in statistics: A review' by A. Antoniadis. *Journal of the Italian Statistical Association*, 6:131–138, 1997.

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.

J. Fan and H. Peng. Nonconcave penalized likelihood with a diverging number of parameters. *Annals of Statistics*, 32(3):928–961, 2004.

J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1:302–332, 2007.

W. J. Fu. Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7:397–416, 1998.

E. I. George. The variable selection problem. *Journal of the American Statistical Association*, 95:1304–1308, 2000.

G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference and prediction*. Springer, New York, 2001.

A. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

D. R. Hunter and R. Li. Variable selection using MM algorithms. *Annals of Statistics*, 33:1617–1642, 2005.

J. Kuha. AIC and BIC: Comparisons of assumptions and performance. *Sociological Methods and Research*, 33:188–229, 2004.

R. Li. Asymptotics and characterizations of nonconvex penalized least squares estimators. Technical report, Department of Statistics, Pennsylvania State University, September 2001.

R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Wiley, Hoboken, NJ, 2nd edition, 2002.

X. Luo, L. A. Stefanski, and D. D. Boos. Tuning variable selection procedures by adding noise. *Technometrics*, 48:165–175, 2006.

C. L. Mallows. Some comments on C_p. *Technometrics*, 15:661–676, 1973.

G. C. McDonald and R. C. Schwing. Instabilities of regression estimates relating air pollution to mortality. *Technometrics*, 15:463–482, 1973.

J. L. Myers and A. D. Well. *Research design and statistical analysis (2nd ed.)*. Lawrence Erlbaum Associates., 2003.

J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman. *Applied linear statistical models*. Richard D. Irwin, Inc, Homewood, Ill., 4 edition, 1996.

G. L. E. Nguefack. *Estimating and correcting the effects of model selection uncertainty*. PhD thesis, University of G ottingen, 2005.

H. Öjelund, H. Madsen, P. J. Brown, and R. Thyregod. Prediction based on mean subset. *Technometrics*, 44:369–378, 2002.

SAS Institute, Inc. *SAS/STAT ®9.1 User's Guide*. SAS Institute, Inc., Cary, NC, 2004.

T. Sawa. Information criteria for discriminating among alternative regression models. *Econometrica*, 46:1273–1291, 1978.

J. L. Schafer and J. W. Graham. Missing data: our view of the state of the art. *Psychological Methods*, 7:147–177, 2002.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

A. Sen and M. Srivastava. *Regression analysis*. Springer-Verlag, New York, 1990.

J. Shao. An asymptotic theory for linear model selection. *Statistica Sinica*, 7:221–264, 1997.

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

R. Tibshirani. A simple explanation of the Lasso and least angle regression (http://www-stat.stanford.edu/~tibs/lasso/simple.html), 2002.

H. Wang, R. Li, and C. Tsai. Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika*, 94:553–568, 2007.

S. G. West, L. S. Aiken, and J. L. Krull. Experimental personality designs: Analyzing categorical by continuous variable interactions. *Journal of Personality*, 64:1–48, 1996.

Y. Yang. Can the strengths of AIC and BIC be shared? a conflict between model identification and regression estimation. *Biometrika*, 92:937–950, 2005.

Y. Zhang. Variable selection via penalized likelihood and iterative conditional minimization algorithm. Master's thesis, Pennsylvania State University, 2006.

M. Zhao and K. B. Kulasekera. Consistent linear model selection. Technical Report TR2005-01-ZK, Dept. of Mathematical Sciences, Clemson University, 2005.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–29, 2006.

H. Zou and R. Li. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics*, page In press, 2007.

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

W. Zucchini. An introduction to model selection. *Journal of Mathematical Psychology*, 44:41–61, 2000.

A Appendix: Theory and Technical Details

A.1 Introduction to SCAD-Penalized Estimation

The SCAD approach to regression combines model selection and coefficient estimation in a single procedure in hopes of providing a simple and standardized approach to exploring large datasets. In the linear models context, this is done by iteratively solving modified normal equations that minimize not the usual least squares criterion, but a penalized version of this criterion that attempts to control both the number of parameters and the size of these parameters, to avoid overfitting.

Recall that in multiple linear regression, the goal is to create a model for predicting future values of a numerical response y from p predictor variables $x_1, x_2, ..., x_p$. The model is assumed to be of the form

$$y = \beta_0 + \sum_{j=1}^{p} x_j \beta_j + \epsilon \tag{1}$$

where ϵ is an error term and the β's are unknown constants. To fit the model, we estimate the β's by collecting a sample of data and minimizing the squared error criterion or "residual sum squares"

$$\text{RSS} = \sum_{i=1}^{n} (y_i - (\beta_0 + \sum_{j=1}^{p} x_{ij} \beta_j))^2 \tag{2}$$

where n is the number of subjects (cases) in the sample. This least-squares regression is done very easily by SAS PROC REG and countless other computer software packages.

However, one very common problem in practice is that the number p of predictor variables may be inconveniently large. A large model is harder to interpret and may have less precise coefficient estimates than a smaller model. Therefore, it is common to fit various smaller models using different subsets of the predictors used in the full model (1), and then try to choose one which is parsimonious but still fits the data well.

Preferably, the choice of a model should be made carefully, incorporating personal judgment, prior knowledge and experience, and knowledge about the uses to which a model will be put. However, besides these issues which need to be subjectively considered, it is still useful to have a simple criterion that shows how much the data favor a given model. RSS wouldn't work for this, because by definition, comparing the candidate models on RSS itself would always lead to choosing the fullest, most flexible model available, rather than any of its constrained subsets. This is a form of overfitting, in that past a certain point, predictors may be "fitting noise" (modeling random idiosyncracies of the observed sample) rather than generalizable population patterns. Therefore, it is natural to consider putting a penalty or constraint on the size of the model, in order to get a more realistic measure of fit.

Among the classic penalized fit criteria are Mallows' C_p (Mallows 1973; see also George 2000) and the closely related Akaike's Information Criterion (AIC; Akaike 1973) and Generalized Cross-Validation (GCV; see Craven and Wahba 1979, Golub et al. 1979) criteria; as well as the stricter and more parsimonious Bayesian Information Criterion

(BIC; Schwarz 1978).[2] It is very common to use one or more of these criteria in an attempt to choose the "best" subset of the pool of available predictor variables, either by considering all subsets or by following a "stepwise" heuristic. This is supported in, e.g., SAS PROC REG (SAS, 2004).

All of these criteria measure the size of a model by counting the number of coefficients which need to be estimated under this model, which is equivalent to the number of predictors included. In the linear models context, these criteria are roughly equivalent to a penalized least squares criterion

$$\mathrm{RSS}/\sigma_e^2 + \lambda_0 d_M$$

where d_M is the number of predictors in the model and λ_0 is a constant determining the size of the penalty; λ_0 is approximately 2 for AIC and GCV and $\log(n)$ for BIC. Here σ_e^2 is the error variance. This approach assigns a number to each subset of potential predictor variables, and we call the subset with the lowest such number "best." In the C_p/GCV/AIC context, "best" means the subset which is predicted to have the best predictive performance on future data from the same population. In the BIC context "best" means the subset which is the most probable true model under an approximation to a certain Bayesian model selection scenario. These criteria are crude but can be very useful in practice. BIC tends to select smaller models than AIC. For more information on them, see Shao (1997), Zucchini (2000), Burnham and Anderson (2002, 2004), Kuha (2004), or Yang (2005).

Consider the full model (1), i.e., the one containing all d available predictors. Notice that a predictor has an effect in the model if and only if its coefficient is nonzero. Thus, when we delete a predictor from the model (whether this is done a priori; or on the basis of AIC, BIC, etc; or on the basis of not being judged significant using a hypothesis test, etc.) it is the same thing as if we constrained its coefficient to be zero and then reestimated the model under this constraint. The predictors which are chosen to be set to zero are usually those for which the absolute values of the least-squares estimates of the coefficients, under the full model, were small (i.e., not statistically significant at some level).

Thus, to look at it another way, selecting a subset in a data-driven way is roughly like applying a "hard threshold" where the final model estimate for coefficient j is 0 if the size of the initial full-model estimate $|\hat{\beta}_j|$ is less than some threshold T_j.[3] We might get an entirely different model if we observe $|\hat{\beta}_j| = \mathrm{T}_j + .01$ instead of $|\hat{\beta}_j| = \mathrm{T}_j - .01$ (see Figure 2). Thus, although in one sense we are reducing error variance by choosing

[2]**Important**: Note that SAS PROC REG uses the abbreviation BIC to stand for a different Bayesian criterion by Sawa (1978), and the standard Schwarz BIC used here is denoted as SBC in the native SAS procedures.

[3]This characterization is somewhat too simplistic, since in general (unless all predictors are perfectly orthogonal) the deletion or inclusion of a given predictors depends on the deletion or inclusion of other predictors in a complex way. The nature of this dependence differs according to whether we are using a forward, backward, or stepwise selection, best subset selection, or marginal significance testing approach. However, ignoring this dependence and considering the simple thresholding model provides a useful insight about classical variable selection methods: The inclusion or exclusion of a variable is a rather simplistic yes-or-no decision which can be very dependent on small changes in the data.

a smaller model, we are also introducing new error variance related to uncertainty about which submodel is correct.

Compare this kind of behavior to that of a very different approach to dealing with the problem of many competing predictor variables: ridge regression. Ridge regression was proposed by Hoerl and Kennard (1970) and is described further in various texts such as Sen and Srivastava (1990) and Neter et al. (1996). By introducing some bias to the regression coefficients, it prevents the error variance of regression estimates from exploding in adverse situations such as high collinearity or low sample size. Unlike subset selection methods which minimize $\text{RSS} + \lambda d_M$, ridge regression minimizes $\text{RSS} + \lambda \sum_{j=1}^{d} \beta_j^2$. In both cases, the penalty is intended to keep the model from getting too "large" in some sense, but the subset selection penalty is trying to control the dimension (number of free predictors) of the model while the ridge regression penalty is trying to control the size of the included coefficients.

The ridge regression estimator is stable (not likely to be heavily changed by small changes in the data) because it is a continuous function of the original data. On the other hand, the post-subset-selection estimator is unstable because of all the dichotomous decisions that have to be made about the individual coefficients. This shows an advantage of ridge regression over subset selection. However, a disadvantage of ridge regression is that although it shrinks coefficients, it never shrinks them all the way to zero as model selection does; i.e., it always includes all available predictors. Someone desiring a smaller and more easily interpreted model still needs to stay with subset selection instead of ridge regression, despite concerns about instability.

A possible compromise approach, the LASSO (Least-Absolute Shrinkage and Selection Operator) was suggested by Tibshirani (1996). Mathematically, this is similar to ridge regression but penalizes the absolute values, rather than squares, of coefficients. That is, the LASSO criterion is $\text{RSS} + \lambda \sum_{j=1}^{d} |\beta_j|$. This criterion differs from ridge regression in that it is able to set some coefficients to zero (see Tibshirani, 1996, 2002; Fu, 1998, for insights as to why this occurs). That is, it combines model selection (deletion) with model stabilization (shrinkage). The LASSO criterion at first looks very difficult to minimize, but can be tackled using either a specialized "shooting" algorithm (Fu, 1998), a local quadratic approximation algorithm (Fu, 1998; Öjelund et al., 2002; Fan and Li, 2001), or the new and extremely fast LARS algorithm (Efron et al., 2004). Heuristically, the LASSO is midway between subset selection (which bases the penalty for a coefficient only on whether a coefficient is zero or not, regardless of its size) and ridge regression (which uses the squared norm of the coefficient). [4]

LASSO still has some limitations. In terms of the subset of parameters selected, it is not very parsimonious. In many situations it includes more predictors than are necessary. For someone interested only in predictive performance, this is not important; but someone who is also interested in a small and concise model may find that LASSO emphasizes shrinking a little too much and selecting not enough. Furthermore, all coefficients are shrunken, even if they are far from zero and there was no question of whether they should

[4]SAS has an experimental procedure to implement the LASSO, known as GLMSELECT; see `http://support.sas.com/rnd/app/da/glmselect.html`.

be deleted. It might make more sense to limit bias by only shrinking coefficients which are relatively close to zero and letting the most important coefficients stay relatively unchanged, if this could be done without sacrificing continuity. This observation led Fan (1997), Antoniadis and Fan (2001), and Fan and Li (2001) to propose the Smoothly Clipped Absolute Deviation (SCAD) penalty, a variation of the LASSO penalty with more nuanced shrinkage properties.

The shrinkage imposed by a continuous penalty function such as LASSO or ridge depends on its derivative rather than its raw value(see Fan and Li, 2001). Notice that the LASSO function's derivative, for nonzero β_j, is simply λ, indicating that all coefficients are penalized equally. The SCAD penalty is

$$\mathcal{P}(\beta_j) = \begin{cases} \lambda|\beta| & \text{if } 0 \leq |\beta| < \lambda \\ \frac{(a^2-1)\lambda^2-(|\beta|-a\lambda)^2}{2(a-1)} & \text{if } \lambda \leq |\beta| < a\lambda \\ \frac{(a+1)\lambda^2}{2} & \text{if } |\beta| \geq a\lambda \end{cases} \tag{3}$$

and for nonzero β_j it has the following derivative:

$$\dot{\mathcal{P}}(\beta_j) = \begin{cases} \lambda \text{sgn}(\beta_j) & \text{if } |\beta_j| < \lambda \\ (a-1)^{-1}\left(a\lambda - |\beta_j|\right))\text{sgn}(\beta_j) & \text{if } \lambda < |\beta_j| < a\lambda \\ 0 & \text{if } |\beta_j| > a\lambda \end{cases} \tag{4}$$

The SCAD penalty is compared with the ridge and LASSO penalties in Figure 1. It is also compared with a penalty which is zero for zero coefficients and a positive constant for nonzero coefficients; AIC, BIC, etc., can be expressed in terms of such a penalty since they measure model size in terms of a count of nonzero coefficients.

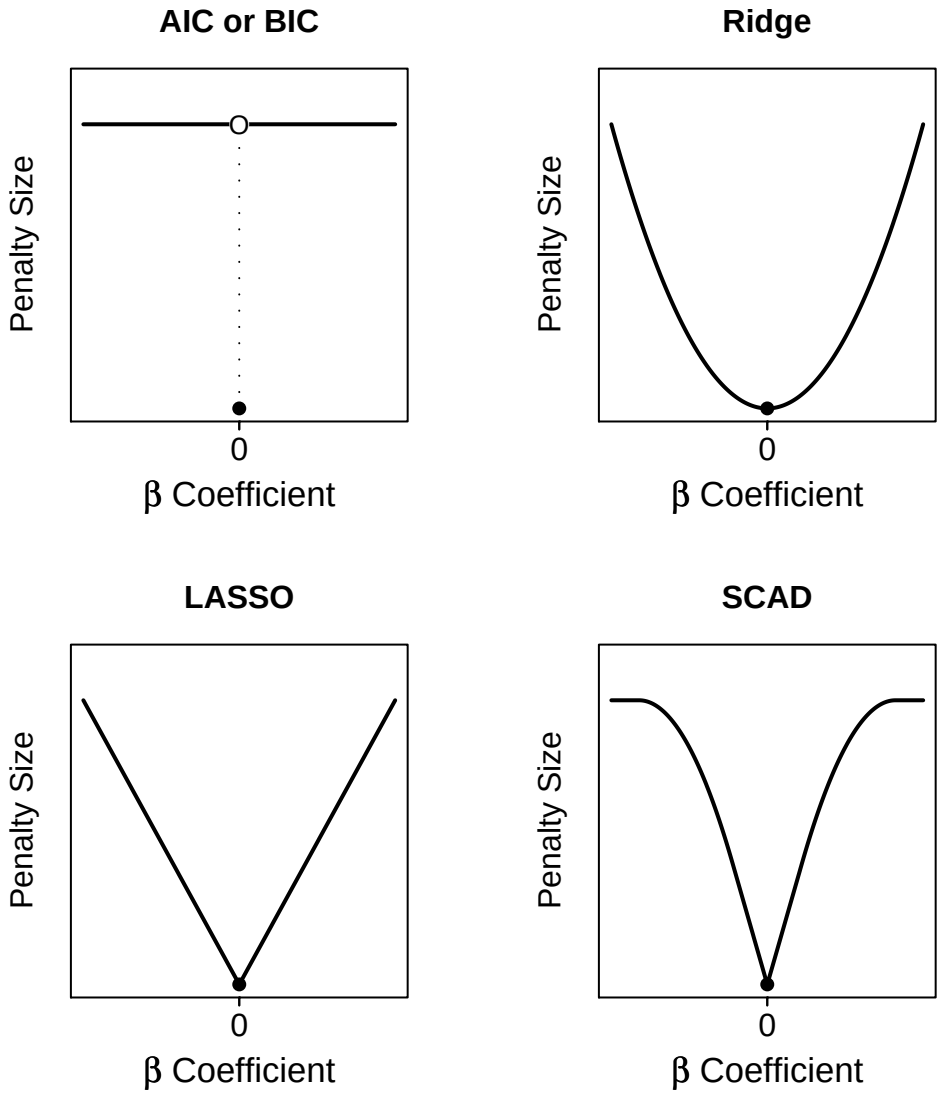
Like the LASSO (Tibshirani, 1996; Efron et al., 2004) and ridge penalties, and unlike the discrete count penalty used by best-subset or stepwise approaches, the SCAD penalty remains continuous, so some shrinkage will be applied and hopefully there will be some added stability. This is somewhat in the spirit of empirical Bayes (although it is not the same, as it is based on geometric and frequentist heuristics rather than Bayesian ones). However, unlike the LASSO and ridge penalties, the SCAD penalty is bounded as a function of β_j (its derivative is zero for large β_j), meaning that it imposes proportionally less bias on large coefficients. SCAD is similar in some ways to the new Adaptive LASSO of (Zou, 2006).

A.2 Properties of the SCAD Estimator

In introducing SCAD for variable selection, Fan and Li (2001) gave two theoretical justifications for a SCAD-penalized estimator. The first is based on comparing SCAD to other methods in a simple case, and the second is based on asymptotics.

To explore the simple case, suppose that each predictor is totally uncorrelated with all other predictors. In this case, each variable is effectively evaluated on its own to determine whether it is significant enough to include in the final model. Significance testing, stepwise testing, and information criteria are all equivalent to a "hard thresholding" rule which includes the variable if the absolute value of its initial coefficient estimate is greater than

Figure 1: Relative Shapes of Penalty Functions



some constant (which depends on the α level of the test or the size of the penalty) and excludes it otherwise (sets its coefficient to zero). Ridge becomes equivalent to a shrinkage rule that multiplies each coefficient estimate by a constant less than one (something like Stein shrinkage); note that this never sets an estimate to zero except in the unlikely case that it was zero already. LASSO and SCAD become rules that combine shrinkage and selection. These rules are shown in Figure 2. Fan and Li (2001) argued that in many cases the SCAD rule was the most desirable of the four for variable selection, since the ridge rule never sets insignificant coefficients to zero, the hard thresholding rule is discontinuous, and the ridge and LASSO rule introduce substantial bias for significant coefficients. This can be seen by comparing the curves in Figure 2 to the dotted diagonal lines representing unpenalized estimates. Departure from the diagonal line represents bias (which is not always bad, but should be minimized, all else being equal).

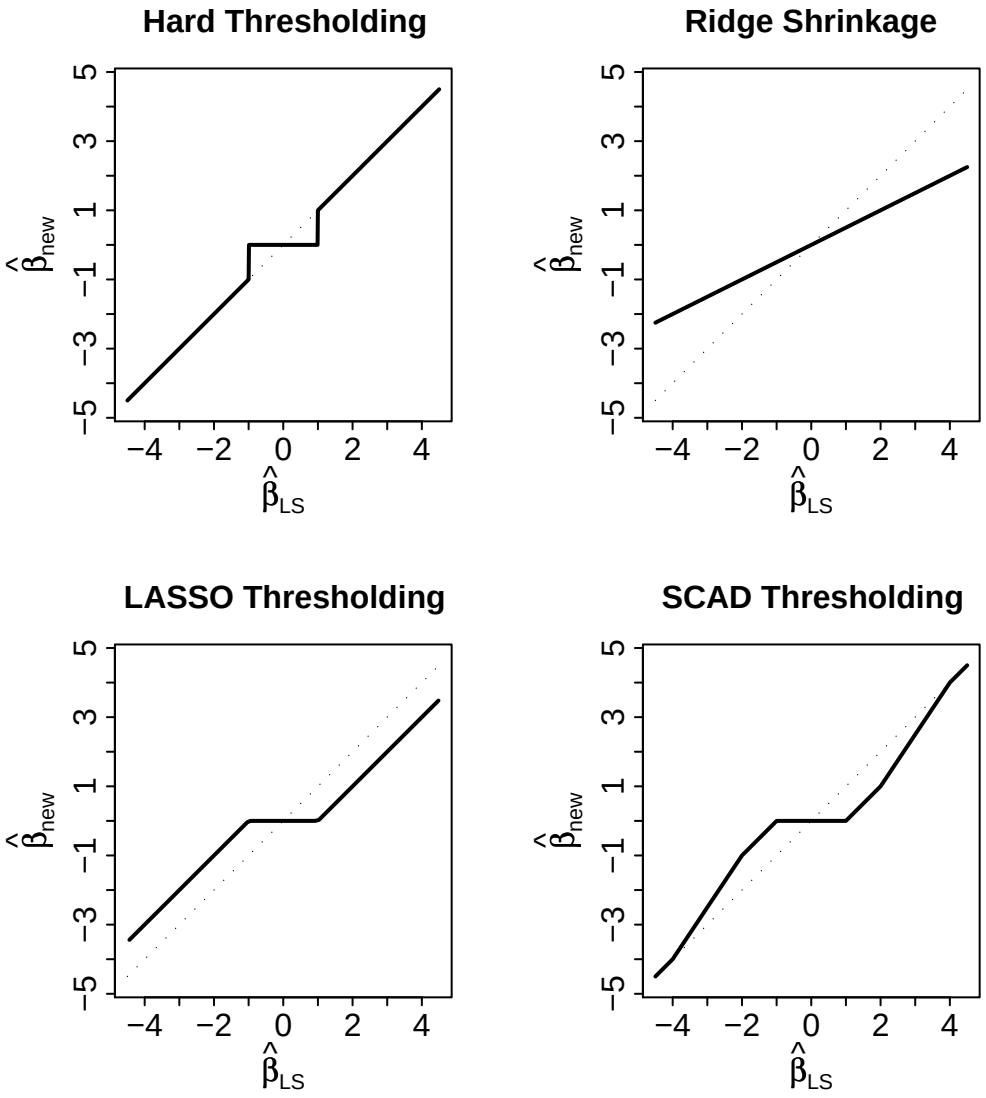
In the more usual correlated-predictors case it is harder to graphically compare the different penalty approaches in a general way, because the importance of particular predictors cannot be considered without knowing what other predictors are present. However, the heuristics from the simple case still provide some insight.

Another way to describe SCAD is in terms of its asymptotic behavior as $n \to \infty$, as described in Li (2001), Fan and Li (2001), Fan and Peng (2004), Zhao and Kulasekera (2005), and Zou and Li (2007). Suppose that the full model (1) is adequate and correct but that some of the coefficients in it are zero. Suppose that the number of coefficients and their true values are fixed, and let $n \to \infty$. If λ is chosen in such a way that $\lambda \to 0$ but $\sqrt{n}\lambda \to \infty$, then the SCAD estimator is consistent for the true subset (i.e., correctly identifies which coefficients are zero and which are nonzero, with probability approaching one), just like BIC subset selection. The bias introduced by shrinkage approaches zero since $\lambda \to 0$. Thus, in a sense the asymptotic variance is the same as if the true subset had been known in advance, a finding known as the "superefficiency" or "oracle" property (Fan and Li, 2001).

The oracle property is an advantage of the SCAD, although not an exclusive one for SCAD alone. The LASSO cannot attain this property in general, except under special conditions. Best-subset with BIC has this property trivially, but can be computationally difficult or impossible if the number of predictors is large. Stepwise selection with BIC may have this property under appropriate assumptions, but stepwise procedures are rather *ad hoc* and often criticized for several reasons. The new adaptive lasso of Zou (2006) has the oracle property and has seems to have quite favorable convergence properties; see also Zou and Li (2007).

Also, the idea of selection consistency or the oracle property has itself been controversial. Of course, it does not really guarantee that the selected model for a given dataset will be the optimal performer in any sense. Because it is based the idea of on selecting the correct subset with probability approaching one as $n \to \infty$, it may not tell us much about finite-n situations where we do not know if we have found the correct subset or not. In fact, for modest n consistent estimators often select too few predictors for optimal future predictive performance (see, e.g., Yang, 2005). Therefore, in our package we allow λ to be selected either using BIC (which corresponds to $\lambda \to 0$ and $\sqrt{n}\lambda \to \infty$ so that the oracle property holds asymptotically) or GCV (which selects a smaller λ so that we do not get

Figure 2: Penalized Estimators as Functions of the Least-Squares Estimate in the Orthogonal-Predictors Case



Note. For each penalty, the x-axis represents the least-squares estimate, and the y-axis represents the final penalized estimate. The curves show the different shrinkage or thresholding rules relating the initial to the final estimate. The dotted diagonal line, shown for comparison, represents a rule in which the final estimate was the initial estimate, i.e., ordinary least-squares with no shrinkage or selection.

the asymptotic oracle property but do have a smaller risk of underfitting), depending on the user's preference.

A.3 Details on the Procedure

There are two main computational issues in obtaining a SCAD estimate: how to find a SCAD estimate given a value of λ, and how to choose a value of λ. Secondary issues include how to find standard errors and how to deal with variables on different scales of measurement. PROC SCADLS can handle all of these issues automatically, but we include this section in order to describe how this is done.

A.3.1 Estimating the Coefficients

The SCAD criterion in the context of linear models is defined as

$$\text{RSS} + \sum_{j=1}^{d} n\mathcal{P}_{\lambda_j}(\beta_j) \tag{5}$$

where $\mathcal{P}(\beta_j)$ equals (3) and λ_j is the tuning parameter applied to β_j.

The usual way to minimize a loss criterion such as (5) is to solve an equation setting its first derivative to zero. This works for convex loss criteria such as quadratically penalized least squares (ridge regression). For instance, in ridge regression the criterion to be minimized is $\text{RSS} + \lambda \sum_{j=1}^{d} \beta_j^2$, or in matrix terms, $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \mathbf{X}^T\mathbf{x}$. This is a convex function of $\boldsymbol{\beta}$ whose minimum is found by solving $-\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta} = \mathbf{0}$, i.e., $\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}$. (If $\lambda = 0$ this is ordinary least squares.)

Minimizing more complicated penalized least squares functions becomes more difficult because they are not necessarily convex and are not everywhere differentiable. For example, with LASSO or SCAD, we would like to solve the equation

$$-2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \sum_{j=1}^{d} \dot{\mathcal{P}}_{\lambda_j}(\beta_j) = \mathbf{0},$$

but we cannot do so explicitly, or even using a Newton-Raphson algorithm directly, mainly because $\dot{\mathcal{P}}_{\lambda_j}(\beta_j)$ is undefined at zero. Therefore, some modified algorithm is needed.

The algorithm proposed by Fan and Li (2001) for minimizing the SCAD function relied on iterative local quadratic approximations to the SCAD function; we simply call this the LQA algorithm. The idea of this modification upon the Newton-Raphson algorithm is as follows: Perform the estimation in steps, with the initial step being the unpenalized (ordinary least-squares) estimate. On a given step, coefficients which are very close to zero (less than a threshold, say 10^{-4}) should be set to zero. For other coefficients, their new values are estimated by approximating the SCAD penalty function for each coefficient by a quadratic function with similar curvature at that coefficient, resulting in a local ridge regression. This approach works rather well; it is described in detail in Fan and Li (2001) and its convergence properties are studied in Hunter and Li (2005).

A newer approach is usually quicker and requires fewer iterations, the Iterative Componentwise Minimization algorithm, is described in Zhang (2006). It works by repeatedly minimizing the SCAD criterion one component at a time, leaving the estimates for the other components fixed, until all estimates converge. It works on the same idea as the coordinate-wise descent algorithm proposed by Friedman et al. (2007) for LASSO. The adjustment to each component is based on a fixed-point characterization derived in Li (2001). The ICM and LQA algorithms give essentially the same answers. LQA is better understood but ICM is very much faster for large datasets because it does not involve repeated matrix inversions.

A.3.2 Selecting λ

SCAD, like other penalized estimation approaches, depends on a "tuning parameter" λ. When $\lambda = 0$, there is no penalty and we are simply fitting ordinary least squares to the full model. If λ is too large then the penalty becomes so strong that all variables are deleted and the model becomes trivial (i.e., $y = \beta_0 + \epsilon$). Somewhere in between, SCAD behaves more desirably. However, since λ is rather abstract, we may not be sure how to choose it.

There are several possible approaches. One approach is to use the fact that, for both LASSO and SCAD, λ is the threshold for model inclusion for the standardized regression coefficient, assuming that all predictors are mutually orthogonal (independent); see Figure 2. As an example, suppose $\lambda = .1$ and $a = 3.7$. The coefficient is set to zero if the initial standardized estimate is less than .1, is shrunken somewhat to maintain continuity if the initial standardized estimate is between .1 and .37, and is left unchanged if the initial standardized estimate is higher than .37. So one might want to set λ to the smallest coefficient value that one would consider practically different from zero. However, this is still rather abstract, and besides, this thresholding characterization does not hold exactly if there is nonzero correlation between the predictor variables, which is almost always the case. Another approach is to try to use asymptotic results. Fan and Li (2001) found that β_j estimates from SCAD are $\sqrt{n}$-consistent as long as $\lambda \to 0$, and have the asymptotic oracle property if $\lambda \to 0$ and $\sqrt{n}\lambda \to \infty$. So we might want a λ on the order of $\sqrt{\log(n)/n}$ or $\sqrt{\sqrt{n}/n}$. This still is not helpful enough, though, because "on the order of" is a very vague term: we don't know whether to use $\frac{1}{5}\sqrt{\log(n)/n}$, $5\sqrt{\log(n)/n}$, etc.

Therefore, a better way to select the tuning parameter is to adapt one of the classical criteria such as GCV or BIC to compare the various candidate SCAD models fitted at each of a list or "grid" of candidate λ values. For instance, we could try $\lambda = .01, .02, .03, ..., .99$, do a SCAD regression for each of these, and choose the resulting model which gives us the best BIC value. The BIC for linear model selection is

$$\text{BIC} = -2\ell(\mathbf{x}_m, \mathbf{y}) + \log(n)d_m = n\log(\text{RSS}_m/n) + \log(n)d_m$$

where $\mathbf{x}_m$ are the predictors in the candidate model, RSS_m is the RSS for the candidate model, and d_m is the size of the candidate model. The use of this criterion as a tuning parameter selector for SCAD was suggested by Wang et al. (2007). A different λ selector

had been originally suggested by Fan and Li (2001), namely a version of the Generalized Cross-Validation (GCV) penalty:

$$\text{GCV} = \frac{\text{RSS}_m}{n(1 - d_m/n)^2}.$$

This is not a very strict penalty. It represents about the same level of shrinkage as AIC, C_p, or leave-one-out cross-validation (see Shao, 1997; Hastie et al., 2001). The tradeoff between SCAD/GCV and SCAD/BIC is very similar to the classic tradeoff between AIC and BIC (see Shao, 1997; Kuha, 2004; Yang, 2005). The former is less likely to delete an actually important variable, and the latter is less likely to include unimportant variables. Neither is universally better than the other, so a user may choose between them based on the goals of a particular data analysis.

The model degrees of freedom, d_m, can also be conceptualized in more than one way. The easiest way is to treat it as a count of how many predictor coefficients are in the model (i.e., are not set to zero). (It does not matter whether we include the intercept in this count or not, since conventionally all models in question will include an intercept unless there is a good reason not to do so.) Such a count does not take shrinkage into account; e.g., it doesn't count a 20% shrunken coefficient as being only .8 of a coefficient. There are various alternative measures of d_m which do measure the effects of shrinkage in addition to selection, in an attempt to avoid choosing an overly constrained model. Fan and Li (2001) suggested a definition based on one widely used in linear smoothing (as in ridge regression and spline fitting), specifically $\hat{d}_m \equiv \text{tr}(\mathbf{P})$ where $\mathbf{P} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + n\mathcal{P}(\hat{\boldsymbol{\beta}}))^{-1}\mathbf{X}^T$. Intuitively, this takes into account how much less variable $\hat{y}$ is in the reduced model as opposed to the full model. By default, PROC SCADLS uses the simple count (`DF COUNT`) to increase computational speed, but the alternative version is also available (`DF PROJ`).

There are other possibilities for choosing tuning parameters, including five- or ten-fold cross-validation (see Fan and Li, 2001) or perhaps temporarily augmenting the model with random data and finding the smallest λ required to delete the bogus data (see Luo et al., 2006). These are not currently supported in PROC SCADLS but could be implemented as macros.

A.3.3 Finding Standard Errors

Fan and Li (2001, p. 1354) showed that an asymptotic covariance estimate for the included coefficients $\boldsymbol{\beta}_m$ of the SCAD-penalized $\boldsymbol{\beta}$ is:

$$\hat{\text{Cov}}(\boldsymbol{\beta}_m) = \left(\mathbf{X}^T\mathbf{X} + n\mathbf{D}(\boldsymbol{\beta})\right)^{-1} \mathbf{C} \left(\mathbf{X}^T\mathbf{X} + n\mathbf{D}(\boldsymbol{\beta})\right)^{-1} \tag{6}$$

where $\mathbf{C} = \hat{\text{Cov}}(\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}))$. $\mathbf{D}$ is a diagonal matrix whose elements are $\dot{\mathcal{P}}(|\beta_j|)/|\beta_j|)$, reflecting the relative shrinkage imposed by the penalty. Once we have $\hat{\text{Cov}}(\boldsymbol{\beta}_m)$ we calculate standard errors in the usual way by taking the square roots of the diagonal elements. In the current version of PROC SCADLS, we use the model-based estimate of $\mathbf{C}$ under homoskedasticity: $\hat{\sigma}^{-2}(\mathbf{X}_m^T\mathbf{X}_m)$, where $\hat{\sigma}^2$ is the variance estimate from the reduced model. ($\mathbf{C}^{-1}$ would be the covariance matrix of $\boldsymbol{\beta}$ if there were no shrinkage.)

These standard error estimates have some limitations: they are asymptotic rather than exact, and they assume that the correct model is chosen with probability approaching one. That is, they effectively condition upon the true model having been chosen, just like standard errors calculated after using a classical variable selection method. Also, like standard errors calculated after ridge regression, they take into account the reduction in sampling variance due to the penalty but do not take into account the increase in bias. Thus, the standard errors are likely to be somewhat too small, especially for higher values of λ. This may not be a huge problem, since the data-driven λ selection methods used here should prevent λ from becoming unreasonably large. However, it may be of interest in the future to explore an alternative technique such as bootstrapping to try to include a more realistic assessment of uncertainty in standard error estimates (although this is not a simple problem; see Nguefack, 2005).

A.3.4 Standardizing the Dataset

The size of the SCAD penalty on a model depends on the β_j estimate for its coefficients, and yet the β_j estimates partially depend on issues other than the practical significance of the predictor (in terms of variance accounted for). For example, they also depend on the scales on which the predictor variables are measured, and the scale on which the response variables are measured. That is, if $y = .003x + \epsilon$ where x is measured in kilometers, then $y \approx 3000x$ if x is re-measured in millimeters. This may lead us to be concerned about whether a penalty of a given size will act arbitrarily. We certainly do not want a selection procedure to consider β_x insignificant in the first case but significant in the second, since in fact it has the same substantive meaning in both cases. The same problem presents itself for ridge and LASSO estimation as well. The usual solution, and the one which we follow here, is to temporarily standardize all of the predictors so that they are all on comparable scales of measurement. The penalized estimation procedure is then applied, and then the final estimates are back-transformed to the original scale of measurement.

Specifically, all of the predictor variables, as well as the response variable, are standardized to have a mean of zero and a variance of one, by subtracting their original means and dividing by their original standard deviations, e.g.: $y^* = (y - \bar{y})/s_y$ where $\bar{y}$ is the mean of y and $s_y = \sqrt{(\sum_i (y_i - \bar{y})^2)/n}$. Then we find the standardized coefficient estimates $\hat{\beta}^*$. To return them to their original scale of measurement for the final output, we multiply each coefficient β_j, as well as its standard error, by s_y/s_{x_j}. The standardized model has zero intercept, so to get the new intercept estimate we must calculate $\bar{y} - \sum_j (\sigma_y/\sigma_{x_j})\hat{\beta}_j x_j$. The intercept standard error under the final model is calculated as $n^{-1}\sigma_e\sigma_y\sqrt{\mathbf{t}^T\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}})\mathbf{t}}$ where σ_e is the error standard deviation in the standardized model and $t_j = s_y\bar{x}_j/s_{x_j}$.

B Appendix: Examples

In this section we work through some examples that show how PROC SCADLS works in various situations, and compare some different approaches.

B.1 Small Simulated Dataset

As a simple first example of using SCAD, we analyze an artificial dataset very similar to those used in some of the Monte Carlo studies in Tibshirani (1996) and Fan and Li (2001). There are 100 cases (i.e. "subjects"), each with a single response variable and eight predictor variables. The predictor variables, x_1 through x_8, were all generated as independent standard normals. The responses y had previously been generated as $5 + 3x_1 + 2x_2 + 1.5x_5 + 3\epsilon$ where the ϵ's were independent standard normal. First let us read in the data.

```
DATA TEST1;
 INPUT y x1 x2 x3 x4 x5 x6 x7 x8;
   DATALINES;
    0.7135 -1.0626  0.7590 -0.6732  0.8885 -0.9389 -0.3818  0.1127  1.0560
    1.6676 -1.5116  1.4317  0.7694 -0.1712 -0.0970 -1.4981 -0.4805  1.1329
    1.0240 -0.9424  0.4283 -1.3002  0.6369 -1.2380  0.1444  1.4361  0.2710
   12.1233  0.8519  1.7792  0.0462 -0.0378  0.1808  0.4751 -0.3108  1.3437
    7.7042  0.6245 -0.8777 -0.6578 -0.0283  0.4522 -0.6585 -0.1024 -0.2741
   10.4535  1.2777  0.1095 -0.4458 -0.3755 -0.1439 -0.6030 -0.0410 -1.2973
    2.9341  0.1941 -1.0241 -0.5028 -0.6683 -0.5703 -0.3055  0.2303 -0.3262
    4.8074 -0.5370 -0.0474  0.6498  0.2650  0.7410 -1.9529  0.4793  0.2360
    9.9254  1.1901  1.1130 -1.5731  0.5149  0.8707  0.0406  0.1151  0.2023
    2.4488 -0.8418 -0.5804  1.1765  0.3028  0.3532  1.1155  0.4236 -0.8443
    7.2965  0.3089  0.3888  0.7393 -1.3369  0.8352 -0.1143  0.6764 -1.7243
    9.5397  0.8709 -0.6241  0.7710  0.7063  1.2302  0.1131  0.9813 -0.5454
    2.8196  0.6158 -2.9541  0.8880  1.1943  1.2662  0.8657  0.8557 -0.4251
   -1.0005 -0.1167 -0.3240  2.0873  0.6128 -0.9921  0.3428 -0.8410  1.4691
    7.9809  0.2214 -1.7367 -3.1569 -1.4121  0.5983 -0.9148  0.1365  0.4053
    9.7050  0.4596  2.4290 -0.0668  0.7210 -0.8176  1.7697 -0.5697  0.2593
    3.5988  0.7408 -0.0755  1.8771  0.1477 -1.5362 -1.7069 -1.3522  0.5833
    3.8925 -0.4747 -0.2618 -0.2418 -2.7012  0.1760  0.2338  0.4839  0.5546
    7.5628 -0.4249  1.4401 -0.3175  0.7627  0.4810  1.1566  2.0936 -0.7501
    4.4317 -1.3722  0.5012  1.0261  0.9193  1.1756  0.5730 -0.8774  0.1826
    8.7069  1.1339 -0.6055  0.6635 -1.0072  0.7144 -0.1383  0.2092  0.3003
    8.8763  0.7029  0.0547  1.3407 -1.9140  0.0401 -0.5720 -0.1978  0.3196
    8.4676 -0.8994  2.4867  0.5015 -1.6870 -0.6123  0.2437 -0.7146 -0.3448
    3.9714 -1.4022  0.6884  1.1130 -1.6294  0.9710  0.6218 -2.2792  0.7932
    3.6611 -0.2143 -0.8850 -0.5643  1.2007 -0.4501 -1.4589  1.7898  0.9685
   10.6914  0.8751  0.6378  0.2530  0.6907  1.3081 -0.6095  0.5391 -0.4177
    7.3058  0.0496  0.9085 -0.2919  0.4941 -0.4262  0.7294 -2.0759  0.2131
    2.1085 -0.4869 -0.8088  1.2236  1.4163 -0.9907 -1.3140  1.5296  0.5896
    7.2639  0.3131 -0.4922 -0.4604 -0.6687  0.8315 -0.3339  0.3616 -0.1142
   -0.7601 -0.8550  0.4753  1.0664 -2.2903 -1.0365 -0.5631  2.2976  0.4530
    6.1289  0.2710  0.4798  0.5140 -1.3196  0.6678  1.7402 -0.0935  0.4349
    7.1117  0.2236 -0.7204 -0.0761 -0.2875  0.2879 -0.3132  0.5311 -0.5560
    5.9097 -0.3758  0.0119 -0.1419  0.4097  0.1097  1.4533  0.6773 -0.4948
    0.8069 -1.2365 -0.7831  1.7836  0.0918 -0.9617 -1.4398 -0.1227  0.0991
   10.1951  0.8695  1.9774 -1.9033  0.3189 -1.0485 -0.2177  1.2549  0.2896
   10.3685 -0.1183 -0.2279  0.9195 -0.6634  0.8604 -1.0162 -0.4512 -0.6612
   -0.0566 -2.6392  0.6756 -0.2228 -0.1525 -0.5456 -1.0072 -1.3255  0.3410
    6.4738  0.1283 -0.8861  0.4017  0.0245  0.0279  1.1144  1.1298 -0.2375
    5.1363  0.2388  0.7520  0.6968 -2.1382 -1.6991 -1.4319  0.4215  1.9272
    5.1573  1.1401 -1.0024  0.3093 -0.5689 -0.5421  0.0841 -0.4780 -0.0308
   12.5305  1.3650  0.2778  0.0569 -1.2677  0.6055 -0.2264 -0.4799 -0.4118
   -1.0739 -2.1471  0.5810  0.2213 -3.3217  1.0252 -0.6118  0.5540  0.7138
```

```
   2.8193  0.4891 -2.0960  0.4425 -1.9448  2.0732  0.6397 -1.1022 -0.6301
   4.7387 -1.5336 -0.8889  0.9711 -0.3332  1.0037 -0.9170  0.2304 -1.2971
   2.8625 -0.3828  0.8936 -1.3562  1.0763 -0.1819  0.4188  1.2675 -2.3801
   4.4254 -0.3550  0.7997 -0.4148  1.3026 -1.0663  0.5679  0.1202 -1.8199
   7.5020 -0.1921  1.9600  0.9926  0.5288  0.6359 -0.0593  0.3116  1.4544
  11.7215  1.1067  0.7285  1.1325  2.1342  0.6199  0.6311 -1.2867 -1.2349
   5.0991  1.1369 -0.2187  0.1450 -0.6110 -1.7179 -0.1743  1.4626  0.5266
   3.1756  0.1448 -1.0334 -0.8347 -0.8007 -0.9029 -0.5233  0.0343 -0.5376
  13.7059  1.1154  0.6908  1.1032  1.1591  0.1371  1.3522 -2.4365  0.1734
   7.8078  0.6885 -0.7856  0.3856  0.8703 -0.4437  0.6484 -0.0693 -1.7423
   6.4489 -0.8585  0.2085 -0.7029 -0.0306  1.5785  0.6973 -0.2920 -0.0700
   2.8715 -0.6887 -0.8535 -0.5350  1.3426  0.1206  1.0839  0.9932  0.7569
   6.4145 -0.0735  1.0111  0.8358 -1.2104  0.1119 -1.5481  0.8058  0.5862
   1.7523 -0.1634 -1.3239 -0.3268 -0.2394 -1.1695 -0.8643  0.1083 -0.3643
   4.2625  0.1827 -1.2443  0.0007 -1.6112  1.2631 -0.3010 -0.2698 -0.2530
   6.4286  1.1445 -0.3804 -0.4990  0.6786 -0.6171  0.2420  0.9520 -0.2500
  11.1319  1.5436  1.0161 -0.1367 -0.1249 -0.9149  0.5802  1.0592 -0.1255
   1.0387 -1.1467 -0.3337  0.1028  1.1466  0.1200  1.2535 -0.5845 -0.0719
   1.5294 -0.3757  0.9953 -0.5973 -0.7756 -1.9783 -0.2890  1.4886  0.2820
   7.3268 -0.4339  0.3431  0.6552  0.8101  1.6342 -1.2046  0.5420 -0.8501
  12.4619  2.5107  0.4966  0.5703 -0.6149  0.3539 -1.0171 -0.6225  1.1046
   3.8198 -0.4600  0.1198 -0.4464 -1.6331 -0.2764  0.9132  0.4207 -0.4041
  12.5353  1.1069  0.3674 -0.2788 -0.3438  1.7512  0.8047 -1.2377  0.6840
   6.8439 -0.1330  0.0404  2.0114  0.1527  1.4407 -0.1090  0.1291  1.2037
   7.8090 -0.8821  2.8766 -0.0223  1.4393 -1.2069  0.6867 -0.0225  0.6936
  11.3768  0.0392  1.4040 -1.4690  1.2704  0.8572 -0.6010 -0.2751 -0.6815
   5.9002  0.7692 -0.9911 -0.5900  0.2298  0.1806  1.5989 -1.0862 -0.3780
   8.2382 -0.0078  1.2598  0.8951  0.3275  0.4673  0.6951  0.3369 -0.9805
  -0.7959 -0.9858 -1.3149  0.2003 -0.8155 -0.2580 -0.6294 -0.1811  1.1912
   2.7515 -0.6723 -0.0382 -0.2616  0.3719  1.2595  0.0359  1.2041  0.6825
  13.8757  2.4139  0.1373 -0.0427  0.5078  0.9273  0.3351  1.3260  1.9798
   8.8781  0.7097  1.1434 -0.4395  0.5210 -0.1541  0.2855  0.3089 -0.2960
   8.4822  1.4314 -0.3777 -0.7005 -1.4219  0.6227 -0.7372  0.3194 -1.9467
   9.0336  0.5849  0.6207  0.6262 -1.2908  0.2297 -1.0295  0.7398  0.9746
   6.5039 -1.6202  1.5300 -0.4649  0.8719  0.5842 -2.4178  1.7615 -0.8973
   4.1844  0.3487 -0.1930 -3.0113  0.6898 -0.4205  0.3583  0.3950  0.2434
   1.8553 -0.0862 -0.2167 -0.7737 -0.4821  0.2703 -0.3782  0.9268 -1.7011
   7.9075 -0.3837  0.9580 -0.2708  0.0888  1.4762 -0.5513 -1.2724 -0.7375
   1.4299 -1.0881  2.0633  0.2725 -0.2703 -0.7278 -1.2500  1.2916 -0.0867
  -0.1599 -1.9856  0.7816 -1.9161 -1.2548 -0.4580  0.3558 -0.0284 -0.4686
   3.7192  0.6263  0.8962  2.3718  1.0978 -2.5220 -1.0866 -0.6175 -0.5285
   1.6896 -0.1886 -0.1465  0.4465  0.7010 -0.2165 -0.1150 -2.0940  0.2174
   8.3053 -0.4976  2.3882  0.6872 -2.0725 -0.6883 -0.3028  0.8049 -0.1015
  -3.9688 -2.6344  0.9589 -0.0316  0.3501 -0.3845 -0.1773  0.1515  1.2669
   4.6400 -0.9804  0.3121  0.2433  1.9599  1.1448 -0.9926  0.4062  1.3361
  -0.5186 -0.5659  0.0639 -0.0368  0.0569 -1.0248 -0.4063  0.0586  2.5596
   8.1078 -0.3749  0.2981 -0.7809  1.3539 -0.1540  1.1428 -0.1208 -0.6416
   5.9799  0.7379 -0.7322  0.3472 -1.9320  0.8693  0.4803  0.1669  1.5743
   5.1503  0.7609  0.2499 -1.9909 -0.5814  1.5311  0.2913  0.3227  0.6269
  10.7930  0.3388  1.9315 -1.5486  2.0023  0.6405 -1.8998  1.1840  1.6101
   4.4160  0.6029 -1.1616  1.1985 -0.9718 -0.8225  0.8565 -0.3973  0.8252
   2.9410  0.0121 -0.1496 -0.3402  1.1943  0.3447 -0.9382 -0.1168 -0.7922
   1.1663 -0.7261 -0.8211 -1.1442  0.0781  0.0445  0.6717 -0.6126 -0.0064
   5.8549  0.2827  1.8150  0.2671 -0.6633 -0.8039  0.4805  2.7130 -2.2495
   3.2077 -0.2873 -1.1953 -0.0677 -1.7522  1.5334  1.7652 -1.0877  1.3030
   6.6556  1.0907  0.5702 -0.3114 -0.9025  0.3821  1.6328 -1.0260 -1.2070
  -2.4938 -1.8878 -1.0900  0.0833  1.0245  0.4143 -1.2575 -1.6563  0.9900
   0.5449 -0.2085 -0.5811 -0.9247  1.1550 -0.4493  0.4330  0.5511  0.0157
   ;
RUN;
```

Now let us run the SCAD algorithm to fit a parsimonious model to this dataset.

```
PROC SCADLS DATA=test1 ;
   MODEL  y = x1 x2 x3 x4 x5 x6 x7 x8;
```

```
   SELECTION BIC;
   ALGORITHM ICM;
RUN;
```

PROC SCADLS returns the following output.

```
Number of observations:                100
Number of predictors:                    8

Tuning parameter details:

Minimum candidate lambda:             0.0093
Maximum candidate lambda:             1.9953
Selected lambda:                      0.2884
Simple df:                               4

BIC:                                419.2536

Converged in 10 iterations.

Final Estimates:

  Variable      Beta   Std. Errs.
-----------------------------------
Intercept :   5.166721  0.192210
x1        :   2.916775  0.194056
x2        :   1.727906  0.182481
x5        :   1.502878  0.208640
```

Suppose we change `SELECTION BIC` to `SELECTION GCV`. In this case the subset selected remains the same (in this case!). Also, if we change `ALGORITHM ICM` to `ALGORITHM LQA`, many more iterations are required, but the computations still take only a second or less, and we get the same answer.

B.2 Small Dataset Continued: Comparing SCAD with Other Methods

For comparison, let us fit the full model using the ordinary regression procedure which is a built-in part of SAS/STAT®.

```
PROC REG DATA=test1;
MODEL y = x1 x2 x3 x4 x5 x6 x7 x8;
RUN;
```

Part of the resulting output is this table of coefficient estimates.

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	5.21931	0.19485	26.79	<.0001
x1	1	2.92668	0.19675	14.88	<.0001
x2	1	1.72692	0.18341	9.42	<.0001
x3	1	0.02287	0.19900	0.11	0.9087
x4	1	0.18813	0.16970	1.11	0.2705
x5	1	1.48036	0.21212	6.98	<.0001
x6	1	-0.21366	0.21595	-0.99	0.3251
x7	1	-0.23276	0.20647	-1.13	0.2626
x8	1	-0.34015	0.20322	-1.67	0.0976

Also for comparison, let us also run the ordinary regression on the three-predictor model which SCAD chose (and which happens to be the correct data-generating model for the simulation).

```
PROC REG DATA=test1;
MODEL y = x1 x2 x5 ;
RUN;
```

We now see the estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	5.16672	0.19221	26.88	<.0001
x1	1	2.91677	0.19406	15.03	<.0001
x2	1	1.72791	0.18248	9.47	<.0001
x5	1	1.50288	0.20864	7.20	<.0001

In this simple situation the SCAD-penalized estimate was not very different from the estimate obtained by fitting the three statistically significant predictors from marginal hypothesis testing after an ordinary regression.

One might also experiment with a best-subset approach:

```
PROC REG DATA=test1;
MODEL y =  x1 x2 x3 x4 x5 x6 x7 x8
/ SELECTION = RSQUARE AIC BIC BEST=5;
RUN;
```

In this case we get the following output:

Number in Model	R-Square	AIC	BIC	Variables in Model

```
1    0.4960   206.2594   205.5319  x1
1    0.1001   264.2264   262.1647  x2
1    0.0945   264.8391   262.7664  x5
1    0.0222   272.5284   270.3221  x8
1    0.0144   273.3149   271.0954  x6
---------------------------------------
2    0.6532   170.8599   170.6792  x1 x2
2    0.5647   193.6077   192.3895  x1 x5
2    0.5034   206.7635   205.0178  x1 x4
2    0.5032   206.8161   205.0684  x1 x8
2    0.4983   207.7863   206.0018  x1 x7
---------------------------------------
3    0.7749   129.6504   131.9335  x1 x2 x5
3    0.6617   170.3891   169.7090  x1 x2 x8
3    0.6615   170.4531   169.7689  x1 x2 x7
3    0.6552   172.2842   171.4842  x1 x2 x4
3    0.6533   172.8335   171.9990  x1 x2 x6
---------------------------------------
4    0.7803   129.2219   131.8261  x1 x2 x5 x8
4    0.7784   130.0921   132.6065  x1 x2 x4 x5
4    0.7765   130.9286   133.3570  x1 x2 x5 x7
4    0.7756   131.3334   133.7204  x1 x2 x5 x6
4    0.7750   131.6159   133.9740  x1 x2 x3 x5
---------------------------------------
5    0.7831   129.9260   132.8149  x1 x2 x4 x5 x8
5    0.7827   130.1177   132.9826  x1 x2 x5 x7 x8
5    0.7816   130.6154   133.4178  x1 x2 x5 x6 x8
5    0.7806   131.0939   133.8365  x1 x2 x3 x5 x8
5    0.7799   131.3835   134.0899  x1 x2 x4 x5 x7
---------------------------------------
6    0.7854   130.8578   134.0547  x1 x2 x4 x5 x7 x8
6    0.7850   131.0711   134.2363  x1 x2 x5 x6 x7 x8
6    0.7846   131.2618   134.3988  x1 x2 x4 x5 x6 x8
6    0.7835   131.7319   134.7995  x1 x2 x3 x4 x5 x8
6    0.7828   132.0928   135.1072  x1 x2 x3 x5 x7 x8
---------------------------------------
7    0.7878   131.7428   135.3032  x1 x2 x4 x5 x6 x7 x8
7    0.7856   132.7983   136.1784  x1 x2 x3 x4 x5 x7 x8
7    0.7850   133.0698   136.4038  x1 x2 x3 x5 x6 x7 x8
7    0.7849   133.1152   136.4414  x1 x2 x3 x4 x5 x6 x8
7    0.7813   134.7606   137.8092  x1 x2 x3 x4 x5 x6 x7
---------------------------------------
8    0.7878   133.7283   137.4890  x1 x2 x3 x4 x5 x6 x7 x8
```

In this case the best-AIC and best-BIC models actually overfit (they include x_1, x_2, x_5, but also x_8). (This is not a general rule; there will be many cases where SCAD overfits and best-subset does not, and other cases in which one or more methods underfit.)

This example does show one advantage of best-subset over other methods (including significance testing, stepwise procedures as well as LASSO and SCAD). Namely, in best-subsets output like the above we can easily see some of the different subsets available for a given size and judge whether or not they differ greatly in AIC or BIC values. If they do not, we know that our chosen model should not be interpreted too confidently, as though it was the only one available; in other words, we get to see whether there are multiple good models which predict the observed data almost as well as the putative best model, and we may wish to choose one or more of them subjectively based on their theoretical meaning. (Of course this is less computationally feasible in situations with hundreds of predictors; in this case we are stuck with either stepwise or some kind of penalized least squares method like SCAD, LASSO, ridge regression, etc.) Other methods, including LASSO and SCAD including stepwise methods, only provide one subset of any given size, which is more convenient but less informative. How important this limitation is, depends on the goals of the individual investigator. One might try SCAD along with various other methods and compare the results.

B.3 Larger Simulated Dataset

To explore how SCAD works with datasets of nontrivial size, we simulate a problem where there are 200 predictor variables and 1000 observations. The data-generating model still has the form of (1). However, now $\beta_0 = 100$, $\sigma = 5$, and the other β's are [0,1,0,-1,0,1,0,-1,0,...,-1]. The x's are all binary (zero-one) variables, independent among subjects but correlated within subjects such that $\text{Corr}(x_{ij}, x_{ik}) \approx .5$. The errors are independent $N(0, \sigma^2)$. In this challenging problem, there are 2^{200} (about a million billion billion billion billion billion billion) possible submodels and only 1000 data points to evaluate them with, so finding the exact true subset is not a realistic goal. However, we want a procedure which will minimize the number of wrong inclusions (here, odd-numbered predictors included in the model) and wrong deletions (here, even-numbered predictors deleted from the model), i.e., to have good sensitivity and specificity. Because the individual effects are small in ratio to error, and the standard errors are large, it is difficult to detect them all. However, because there are 100 inactive predictors, all of them correlated with active predictors, it is difficult to exclude them all. Thus a model selection method of some kind is needed. 200 predictors may make the dataset too big for an exhaustive best-subsets search to be feasible. SCAD was designed as an alternative method for attempting to handle large datasets like this.

```
proc scadls data=big;
   model y = var1-var200  ;
   selection BIC;
   df COUNT;
   algorithm ICM;
run;
```

For SCAD with BIC, the default MAXITER was not enough; it had to be increased, and finally the procedure took 1303 iterations (a few minutes) to run. SCAD with BIC

falsely excluded 33 of the active predictors (7, 18, 22, 26, 42, 54, 56, 64, 68, 70, 86, 90, 92, 94, 96, 112, 114, 116, 130, 134, 138, 140, 142, 146, 152, 156, 168, 172, 176, 180, 188, 190, 194) and correctly included the other 67. It falsely included 10 of the inactive predictors (15, 69, 81, 83, 89, 111, 127, 139, 143, 167) and correctly excluded the other 90. Of the correct inclusions, all beta estimates had the correct sign.

For SCAD with BIC, 3109 ICM iterations were needed. SCAD with GCV falsely excluded 25 predictors (numbers 18, 22, 26, 28, 42, 56, 64, 66, 68, 90, 92, 94, 96, 114, 116, 130, 134, 140, 142, 156, 168, 176, 188, 190, 194) but included the other 75. It falsely included 37 of the inactive predictors (1, 9, 11, 19, 21, 25, 27, 31, 39, 49, 51, 57, 69, 79, 81, 83, 89, 95, 103, 109, 111, 127, 135, 143, 145, 147, 151, 161, 165, 167, 177, 179, 181, 183, 185, 187, 199) but excluded the other 63. Again, of the correct inclusions, all beta had the correct sign (none were negative).

Which performance was better is obviously a tradeoff that depends on one's relative amount of concern about false inclusions (which are something like Type One errors and make the model more complicated) versus false exclusions (which increase the model bias by constraining nonzero parameters to zero).

B.4 Pollution Dataset

In this example we analyze the pollution dataset McDonald and Schwing (1973) from StatLib (`http://lib.stat.cmu.edu/datasets/pollution`). The goal is to predict city-specific mortality rates per 100000 people (MORT) for a sample of American cities, from a number of predictors including precipitation rates (PREC), average winter temperature (JANT), average summer temperature (JULT), percent aged over 65 (OVR65), average household size (POPN), median education level (EDUC), percent adequate housing (HOUS), population per square mile (DENS), percent non-white (NONW), percent white-collar jobs (WWDRK), percent households with incomes under the poverty line (POOR), nitric oxide pollution level (NOX), sulfur oxide pollution level (SOX), and average humidity (HUMID). We leave out a hydrocarbon pollution variable, HC, that turned out to be almost perfectly correlated with NOX; also, we apply (natural) log-transformations to DENS, NOX, and SOX, which on their original scale are strongly positively skewed. To run SCAD with the GCV tuner we can use

```
PROC SCADLS DATA=pollution DETAILS;
   MODEL mort = prec jant jult ovr65 popn educ hous
          logdens nonw wwdrk poor lognox logsox humid ;
   SELECTION GCV;
   DF COUNT;
   ALGORITHM ICM;
   MAXITER 500;
RUN;
```

The output is shown below:

```
Number of observations:                60
```

```
Number of predictors:                   14
Tuning parameter details:

Minimum candidate lambda:              0.2057
Maximum candidate lambda:             41.6280
Selected lambda:                       3.6992
Simple df:                               10

GCV:                                1121.3422

Converged in 309 iterations.

Final Estimates:

  Variable      Beta   Std. Errs.
-----------------------------------
Intercept :  1933.7641 347.84825
prec      :   2.682706  0.758232
jant      :  -2.592863  0.589249
jult      :  -3.154923  1.664826
ovr65     :  -13.76543  6.727901
popn      :  -148.8091 57.462948
educ      :  -20.47393  6.739721
nonw      :   4.154444  0.991648
poor      :  -33.95322 14.475614
lognox    :  45.320582 12.696981
```

For BIC we get

```
Number of observations:                 60
Number of predictors:                   14
Tuning parameter details:

Minimum candidate lambda:              0.2057
Maximum candidate lambda:             41.6280
Selected lambda:                       5.5291
Simple df:                                7

BIC:                                 608.4714

Converged in 237 iterations.

Final Estimates:
```

```
  Variable      Beta   Std. Errs.
-----------------------------------
Intercept :  857.55407 43.118742
prec      :   2.504568  0.599153
jant      :  -2.067983  0.494621
educ      :  -1.713634  2.416767
nonw      :   3.748743  0.648163
poor      : -22.70902 12.768139
lognox    :  41.454661 12.112746
```

The estimates for the full model, for best-subset selection, and for SCAD selection, are summarized in the following table.

	Full		Best AIC		Best BIC		SCAD, GCV		SCAD, BIC	
	β	SE	β	SE	β	SE	β	SE	β	SE
(Intercept)	1786.90	433.31	1933.76	347.85	812.60	29.43	1933.76	347.85	857.55	43.12
prec	2.58	0.82	2.68	0.76	2.95	0.57	2.68	0.76	2.50	0.60
jant	-2.26	0.69	-2.59	0.59	-2.28	0.49	-2.59	0.59	-2.07	0.49
jult	-3.38	1.99	-3.15	1.66			-3.15	1.66		
ovr65	-13.89	7.26	-13.77	6.73			-13.77	6.73		
popn	-132.50	63.95	-148.81	57.46			-148.81	57.46		
educ	-14.60	9.94	-20.47	6.74			-20.47	6.74	-1.71	2.42
hous	-1.13	1.27								
logdens	16.68	14.99								
nonw	3.86	1.11	4.15	0.99	3.76	0.66	4.15	0.99	3.75	0.65
wwdrk	-0.19	1.43								
poor	-34.68	15.36	-33.95	14.48			-33.95	14.48	-22.71	12.77
lognox	42.21	14.81	45.32	12.70	21.67	4.41	45.32	12.70	41.45	12.11
logsox	0.06	0.11								
humid	0.21	0.97								

We can see that best-AIC and SCAD-GCV worked similarly, and best-BIC and SCAD-BIC worked similarly. All of the methods agreed on some variables (such as jant and jult but disagreed on others). One might ask whether these variables "should" be included in a "true" model or not. In reality (outside of simulations), it is unlikely that any observable variable has a partial correlation of exactly zero with the outcome, so in a sense even the full model is nowhere near flexible enough to be "true." In a more practical sense, the choice between a smaller and larger model is based on an investigator's needs and priorities, and a judgment of which error would be more serious in one's own particular situation: including a relatively unimportant variable or excluding a relatively important one.

Examining the table, one might be worried about the vast disagreements among the models regarding the intercept term, both in the estimates and the standard errors. However, these disagreements actually mean little or nothing, because the intercept parameter (which could be interpreted as the predicted value in the unlikely or impossible case that all of the **included** predictors are zero) has a different meaning for each model. This discrepancy would be less if the predictors had been centered at zero.

Incidentally this dataset also provides a good example of why regression coefficients in an observational study or survey should not be interpreted as causal effects. For instance, in all cases, lower January temperatures are associated with lower mortality.

A naïve analyst would have expected the reverse, because people might die of frostbite if temperatures are too low. Instead, January temperature may be a marker for region of the country, which in turn is related to socioeconomic factors. Also, most of the models include precipitation level but do not include sulfur dioxide level, but we would not conclude that water is toxic but sulfur dioxide is harmless. Rather, some of the variables are probably serving as proxies for unobserved characteristics of the geographic location, and some other variables may be hidden by collinearity with others. In fact, the correlation coefficient of logsox with mort is higher than that of lognox, but the small and haphazard sample and a .69 correlation between lognox and logsox make it impossible to separate their effects.

B.5 Nutrition Dataset with Categorical Predictors

In this section we use the plasma retinol dataset submitted by Therese Stukel on StatLib (see `http://lib.stat.cmu.edu/datasets/Plasma_Retinol`). Suppose our goal is to find a parsimonious model to predict people's blood plasma levels of the antioxidant micronutrient beta-carotene, found in carrots and other colored vegetables. The response is BETAPLASMA, the beta-carotene level in nanograms per millileter. Relevant predictors include AGE in years, SEX (1 for male / 2 for female), SMOKSTAT (1=nonsmoker, 2=former smoker, 3=smoker), QUETELET (a numerical measure of weight status, something like body mass index; actually the ratio of weight to squared height), VITUSE (1=frequent vitamin pill user, 2=occasional user, 3=nonuser), CALORIES consumed per day, FAT grams consumed per day, FIBER consumed per day, ALCOHOL drinks consumed per week, CHOLESTEROL milligrams consumed per day, and BETADIET which is the amount of beta-carotene consumed from food per day in micrograms. Because a model for BETAPLASMA which did not include BETADIET would be strange, we will use the FORCEIN option to make sure that BETADIET will be treated as a covariate of special importance and will not be penalized or deleted.

This dataset differs from the pollution dataset in that it involves some categorical predictors. It would be okay to include SEX as a numerical predictor because it has only two levels. However, SMOKSTAT and VITUSE have three levels each. They are ordinal rather than nominal, so they could be imagined to represent an underlying quantitative dimension, but they are not really on a numerical scale. That is, we cannot trust that an occasional vitamin user is really halfway between a frequent user and a nonuser in any sense, especially because "frequent" and "occasional" may not be precisely defined. We also cannot assume that the difference in health between a former-smoker and a never-smoker is the same as that between a current-smoker and a former-smoker in the same way that 2 minus 1 is the same as 3 minus 2. Therefore we will use the CLASS statement. First let us fit an ordinary regression to the data. We can do this using PROC GLM.

```
PROC GLM;
CLASS sex smokstat vituse;
MODEL betaplasma = age sex smokstat quetelet vituse calories
 fat fiber alcohol cholesterol betadiet / SOLUTION;
```

```
RUN;
```

The results are:

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		184.8972965	B	67.94380472	2.72	0.0069
AGE		0.7001720		0.74301573	0.94	0.3468
SEX	1	-31.4630580	B	31.56618636	-1.00	0.3197
SEX	2	0.0000000	B	.	.	.
SMOKSTAT	1	45.9119033	B	30.81886225	1.49	0.1373
SMOKSTAT	2	40.7690632	B	31.09841805	1.31	0.1909
SMOKSTAT	3	0.0000000	B	.	.	.
QUETELET		-5.9340466		1.62873556	-3.64	0.0003
VITUSE	1	78.4010811	B	23.12631952	3.39	0.0008
VITUSE	2	44.3051936	B	25.25345442	1.75	0.0804
VITUSE	3	0.0000000	B	.	.	.
CALORIES		-0.0289505		0.05071633	-0.57	0.5685
FAT		0.0758120		0.80264713	0.09	0.9248
FIBER		6.1935866		2.81269155	2.20	0.0284
ALCOHOL		0.9808388		1.23318655	0.80	0.4270
CHOLESTEROL		-0.0934808		0.10628837	-0.88	0.3798
BETADIET		0.0164936		0.00743796	2.22	0.0273

"B" here means that a block of terms included a linear dependency and could not be estimated. This is just what should happen, and is not a problem, since the dummy code for the baseline must be constrained to zero. Anyway, it seems that the statistically significant predictors of plasma beta-carotene are vitamin use and dietary beta carotene (two obvious potential direct sources of beta-carotene), plus fiber intake and Quetelet ratio (perhaps markers for general healthy lifestyle and diet).

Let's fit a SCAD-penalized model. Suppose we want to be careful not to delete potentially useful predictors, so we use GCV instead of BIC.

```
PROC SCADLS DATA=carrots;
    CLASS sex smokstat vituse;
    MODEL betaplasma =  age sex smokstat quetelet vituse
              calories fat fiber alcohol cholesterol betadiet;
    FORCEIN betadiet;
    SELECTION gcv;
    MAXITER 5000;
RUN;
```

We see the following output:

```
Number of observations:                315
Number of predictors:                   11
Predictor forced in:                                        BETADIET
Tuning parameter details:

Selected lambda:                      5.9042

GCV:                               28916.1483

Converged in 3427 iterations.

Final Estimates:

  Variable     Level     Frequency   Beta   Std. Errs.
-----------------------------------------------------------
Intercept             :           215.71950 62.665893
AGE                   :            0.403144  0.671943
SEX          1        :      42   -4.399076 12.750052
             2        :     273       (Reference)
SMOKSTAT     1        :     157    6.343082 11.295655
             2        :     115    0.000000  0.000000
             3        :      43       (Reference)
QUETELET              :           -5.710370  1.604427
VITUSE       1        :     122   79.960998 22.138816
             2        :      82   41.087703 23.613611
             3        :     111       (Reference)
FIBER                 :            5.098775  2.062451
CHOLESTERO            :           -0.188034  0.074711
BETADIET              :            0.018059  0.007389
```

Age, sex, smoking status, Quetelet ratio, vitamin use, fiber, cholesterol, and (of course) dietary beta-carotene were kept as predictors. In the case of smoking status use, the category opposite the baseline was included in the model as significantly different from the baseline, but the intermediate category was collapsed with the baseline. In such a situation, and unlike unpenalized regression, the choice of baseline matters in terms of the final predicted values. This is because it is the moderate smokers could be joined with the nonsmokers or heavy smokers, depending on which baseline is chosen. This would lead to different predictions for some individuals. To try another baseline, use the DESCENDING option.

To reverse the choice of baseline, we could have instead used the code:

```
PROC SCADLS DATA=carrots;
    CLASS sex smokstat vituse;
    MODEL betaplasma =  age sex smokstat quetelet vituse
```

```
            calories fat fiber alcohol cholesterol betadiet;
    FORCEIN betadiet;
GRIDSIZE 250; MAXITER 250;
RUN;
```

We get

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



Number of observations:                315
Number of predictors:                   11
Predictor forced in:                                      BETADIET
Tuning parameter details:

Selected lambda:                      6.8096

GCV:                               28647.7094

Converged in 392 iterations.

Final Estimates:

  Variable     Level     Frequency   Beta   Std. Errs.
-----------------------------------------------------
Intercept             :           307.44540 62.537938
AGE                   :            0.318870  0.664231
SMOKSTAT    3         :      43   -31.45941 23.176080
            2         :     115    0.000000  0.000000
            1         :     157       (Reference)
QUETELET              :           -5.974736  1.602367
VITUSE      3         :     111   -72.54134 21.096472
            2         :      82   -16.60578 16.518081
            1         :     122       (Reference)
FIBER                 :            4.855713  2.065164
CHOLESTERO            :           -0.182577  0.073385
BETADIET              :            0.017885  0.007364
```