Least Angle Regression

Tim Hesterberg, Chris Fraley
Insightful Corp., `www.insightful.com/Hesterberg/glars`

This is an abridged version of an SBIR Phase II proposal to NIH, submitted 3 April 2006. The Phase I work described here was supported under NIH SBIR Phase I award 1 R43 GM074313-01. This proposal was successful, with subsequent work supported by NIH SBIR Phase II award 2 R44 GM074313-02.

Project Summary/Abstract

This SBIR project aims to produce superior methods and software for classification and regression when there are many potential predictor variables to choose from. The methods should (1) produce stable results, where small changes in the data do not produce major changes in the variables selected or in model predictions; (2) produce accurate predictions; (3) facilitate scientific interpretation, by selecting a smaller subset of predictors which provide the best predictions; (4) allow continuous and categorical variables; and (5) support linear regression, logistic regression (predicting a binary outcome), survival analysis, and other types of regression.

This project is based on least angle regression, which unifies and provides a fast implementation for a number of modern regression techniques. Least angle regression has great potential, but currently-available software is limited in scope and robustness. The outcome of this project should be software which is more robust and widely applicable.

This software would apply broadly, including to medical diagnosis, detecting cancer, feature selection in microarrays, and modeling patient characteristics like blood pressure.

Phase I work demonstrates feasibility by extending least angle work in three key directions–categorical predictors, logistic regression, and a numerically-accurate implementation.

Phase II goals include extensions to other types of explanatory variables (e.g. polynomial or spline functions, and interactions between variables), to survival and other additional regression models, and to handle missing data and massive data sets.

This proposed software will enable medical researchers to obtain high prediction accuracy, and obtain stable and interpretable results, in high-dimensional situations.

Project Narrative

Predicting outcomes based on covariates, determining which covariates most affect outcomes, and adjusting treatment effects estimates for covariates, are among the most important problems in biostatistics. Prediction and feature selection are particularly difficult when there are more possible features than samples; gene microarrays and protein mass spectrometry are extreme examples of this, producing thousands to millions of measurements per sample. LARS excels at feature selection; the proposed software should enable medical researchers to obtain stable and interpretable models with better prediction accuracy in high-dimensional situations.

2 Specific Aims

"I've got all these variables, but I don't know which ones to use."

Classification and regression problems with large numbers of candidate predictor variables occur in a wide variety of scientific fields, increasingly so as improvements in data collection technologies facilitate collecting large numbers of potential predictors. For example, in microarray analysis large numbers of genes are analyzed, and the number of predictors (genes) typically exceeds the number of observations.

Goals in model selection include:

- accurate predictions,
- interpretable models—determining which predictors are scientifically meaningful,
- stability—small changes in the data should not result in large changes in either the subset of predictors used, the associated coefficients, or the predictions, and
- avoiding bias in hypothesis tests during or after variable selection.

Older methods, such as stepwise regression, all-subsets regression and ridge regression, fall short in one or more of these criteria. Modern procedures such as boosting (Freund and Schapire, 1997) forward stagewise regression (Hastie et al., 2001), and the Lasso (Tibshirani, 1996), improve stability and predictions, but can be slow.

Efron et al. (2004) show that there are strong connections between these modern methods and a new method, *least angle regression*, and that a single fast algorithm can be used to implement all of them. They use the term LARS to collectively refer to least angle regression and the fast implementation of the other methods. LARS is potentially revolutionary, offering interpretable models, stability, accurate predictions, graphical output that shows the key tradeoff in model complexity, and a simple data-based rule for determining the optimal level of complexity that nearly avoids the bias in hypothesis tests.

In Phase I we began the process of converting the original academic LARS software to more widely-usable software, by improving numerical stability, generalizing from linear to logistic regression, supporting factor variables, and by making the software easier to use. We produced and have delivered prototype software to our consultants and other outside testers.

Our goal in Phase II is to produce high-quality software for classification and regression based on LARS for widespread use. Specific aims include:

- generalize to additional nonlinear models, such as other generalized linear models, survival analysis, mixed-effects models, robust regression, and classification
- allow for missing data
- handle massive data sets
- provide a graphical user interface (GUI)
- provide a well-designed interface for non-GUI users
- provide a framework that academic collaborators can extend
- support for other types of types of explanatory variables (e.g. polynomial or spline functions, and interactions between variables)

- provide tools for choosing parameters and assessing significance and stability, e.g. C_p, F statistics, cross validation and bootstrap.

Progress in Phase I was excellent, and there are a number of very interested academic collaborators. *(Some text is omitted from this version of the proposal.)*

This project could have a strong positive impact on our nation's health care. Predicting outcomes based on covariates, and determining which covariates most affect outcomes, are among the most important problems in biostatistics.

3 Background and Significance

3.1 Technical Problem or Opportunity

Ten years ago, when the PI (Hesterberg) first joined Insightful (the statistical software company that develops and sells S-PLUS), he asked Brad Efron (2004 President of the American Statistical Association), what he thought were the most important problems in statistics, that Insightful could try to solve. The expectation was that Efron's answer would relate to the bootstrap, given his status as its inventor. Instead, he named a single problem, *variable selection in regression*. This entails selecting variables from among a set of candidate variables, estimating parameters for those variables, and inference—hypotheses tests, standard errors, and confidence intervals.

It is hard to argue with this assessment. Regression, the problem of estimating a relationship between a response variable and various predictors (explanatory variables, covariates) is of paramount importance in statistics (particularly when we include "classification" problems, where the response variable is categorical). A large fraction of regression problems require some choice of what predictors to use. Efron's work has long been strongly grounded in solving real problems, many of them from biomedical consulting. His answer reflects the importance of variable selection in practical statistical work.

Classical tools for analyzing regression results, such as t statistics for judging the significance of individual predictors, are based on the assumption that the set of predictors is fixed in advance. When instead the set is chosen adaptively, incorporating those variables that give the best fit for a particular set of data, the classical tools are biased. For example, if there are 10 candidate predictors, and we select the single one that gives the best fit, there is about a 40% chance that that variable will be judged significant at the 5% level, when in fact all predictors are independent of the response and each other. Similar bias holds for the F test for comparing two models; it is based on the assumption that the two models are fixed in advance, rather than chosen adaptively.

This bias affects the variable selection process itself. Formal selection procedures such as stepwise regression and all-subsets regression are ultimately based on statistics related to the F statistics for comparing models. Informal selection procedures, in which an analyst picks variables that give a good fit, are similarly affected.

Move forward six years, to 2002; little progress was made—Allan Miller wrote in 2002 in the preface to the second edition of *Subset Selection in Regression* (Miller, 2002):

> What has happened in this field since the first edition was published in 1990?
>
> The short answer is that there has been very little progress. The increase in the speed of computers has been used to apply subset selection to an increasing range of models, linear, nonlinear, generalized linear models, to regression methods which are more robust against outliers than least squares, but we still know very little about the properties of the parameters of the best-fitting models chosen by these methods. From time-to-time simulation studies have been published, e.g. Adams (1990), Hurvich and Tsai (1990), and Roecker (1991), which have shown, for instance, that prediction errors using ordinary least squares are far too small, or that nominal 95% confidence regions only include the true parameter values in perhaps 50% of cases.

Problems arise not only in selecting variables, but also in estimating coefficients for those variables, and producing predictions. The coefficients and predictions are unstable (small changes in the data may result in large changes in the set of variables included in a model and in the corresponding coefficients and predictions) and are biased. Miller (2002) notes:

> As far as estimation of regression coefficients is concerned, there has been essentially no

progress.

Moving forward to the present, we believe that the new technique, least angle regression Efron et al. (2004), and its Lasso and forward stagewise variations, offer strong promise for producing interpretable models, accurate predictions, and approximately unbiased inferences.

The academic community evidently concurs. There is a flurry of activity in the area; a Google Scholar search shows 114 citations of (Efron et al., 2004).

Outside Collaboration

We expect to attract a substantial amount of collaboration during Phase II work, for a number of reasons. This is an area of active research, and we maintain strong ties to the academic community. Much of the academic software for LARS and Lasso has been released as S-PLUS packages (`lasso2` (Lokhorst et al., 1999), `brdgrun` (Fu, 2000), `lars` (Efron and Hastie, 2003)) or R packages (`glmpath` (Park and Hastie, 2006b), `elasticnet` (Zou and Hastie, 2005b), `glasso` (Kim et al., 2005b)). Insightful is working to facilitate the use of R packages in S-PLUS; this is a key feature of the next release of S-PLUS, which enters beta testing in April 2006. Our prototype "`S+GLARS`" library is based in part on `lars` and `glmpath`, runs in both S-PLUS and R, and is released under an open source license, GPL 2.0 (GNU Public License), to allow others to build on the framework we develop. *(Some text is omitted from this version of the proposal.)*

We anticipate spending a substantial amount of time during Phase II working with some of these and other collaborators to adapt their algorithms, and in some cases their software, for use by the wider statistical community.

We feel that this collaborative approach is best in this rapidly-developing field.

Part of our strategy is to provide a framework that others can build on, to encourage the creation of high-quality work in S-PLUS and R, to supplement our own efforts. We plan to create front-end functions for specific areas, such as linear regression, generalized linear models, and cox regression, that other people can plug their own fitting (parameter estimation) routines into. We also plan to create utilities for plotting and other post-estimation analysis, and standards for the structure of objects returned by fitting routines; as long as the fitting routines adhere to those standards, third-party statistical analysts could use their choice of fitting routines, with a common user interface.

Note that software developed in academic settings typically requires substantial work to be suitable for the wider community; it often covers only special cases and requires substantial research to generalize, and lacks the attention to detail needed for wide use. By providing a good framework, much of that detail work need only be done once.

The feedback from these collaborators has also informed the priorities in this proposal; they recommended work on generalized linear models, Cox regression, handling missing data, mixed effects models, group models (factors, splines, etc.), support for elastic net and ridge regression, tools for model selection such as C_p and AIC, as well as various suggestions for the user interface.

Related Work

(Some text is omitted from this version of the proposal.)

3.2 Background

In this section we begin with a preface indicating the significance of the problem, describe a diabetes data set we use as an example, then give background in the form of discussion of various methods and their relative merits.

We write LAR for least angle regression, and LARS to include LAR as well as Lasso or forward stagewise implemented by least angle methods. We use the terms predictors, covariates, explanatory variables, and variables interchangeably (except we use the latter only when it is clear we are discussing explanatory rather than response variables).

The example in this section involves linear regression, but most of the text applies as well to logistic, survival, and other nonlinear regressions in which the predictors are combined linearly, where predictions are some function of $\beta_j x_j$. We note where there are differences between linear regression and the nonlinear cases.

The restriction to linear combinations of predictors is not as limiting as it may appear; the predictors may themselves be nonlinear functions of other variables, e.g. polynomial functions for polynomial regression. This includes some versions of the newer "random forest" techniques, where the predictors are themselves trees, certain piecewise constant functions of underlying explanatory variables.

When discussing methods, we begin with "pure variable selection" methods such as stepwise regression and all-subsets regression that pick predictors, then estimate coefficients for those variables using standard criteria such as least-squares or maximum likelihood. In other words, these methods focus on variable selection, and do nothing special about estimating coefficients. We then move on to ridge regression, which in its usual form does the converse—it is not concerned with variable selection (it uses all candidate predictors), and instead modifies how coefficients are estimated. We then discuss the LASSO method, a variation of ridge regression that modifies coefficient estimation so as to reduce some coefficients to zero, effectively performing variable selection. From there we move to forward stagewise regression, an incremental version of stepwise regression that gives results very similar to the Lasso. Finally we turn to least angle regression, which connects all the methods.

Diabetes Data Set

Before describing the new methods and related older methods, we describe a dataset we use to demonstrate and compare the methods.

Ten predictor variables, consisting of age, sex, body mass index, average blood pressure, and six different blood serum measurements, were obtained for each of 442 diabetes patients, as well as the response, a quantitative measure of disease progression one year after baseline. One goal is to create a model that predicts the response from the predictors; a second is to find a smaller subset of predictors that fits well, suggesting that those variables are important factors in disease progression. Part of this dataset is shown in Table 1.

Stepwise and All-Subsets Regression

We begin our description of various regression methods with stepwise and all-subsets regression, methods which focus on which variables are selected for a model, rather than on how coefficients are estimated once variables are selected.

Forward stepwise regression begins by selecting a single predictor variable which produces the best fit, e.g. the smallest residual sum of squares. Another predictor is then added which produces the best fit in combination with the first, followed by a third which produces the best fit in combination with the first two,

Table 1: Diabetes Study: 442 patients were measured on 10 baseline variables; a prediction model is desired for the response variable, a measure of disease progression one year after baseline.

Patient	Age	Sex	BMI	BP	S1	S2	S3	S4	S5	S6	Y
1	59	2	32.1	101	157	93.2	38	4.0	4.9	87	151
2	48	1	21.6	87	183	103.2	70	3.0	3.9	69	75
3	72	2	30.5	93	156	93.6	41	4.0	4.7	85	141
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
442	36	1	19.6	71	250	133.2	97	3.0	4.6	92	57

and so on. This process continues until some stopping criteria is reached, based e.g. on the number of predictors and lack of improvement in fit. Table 2 shows the first 6 steps of forward stepwise least-squares linear regression applied to the diabetes data set.

Table 2: Forward Stepwise: The single best predictor is BMI; the best predictor in combination with BMI is S5, etc. The sequence is BMI, S5, BP, S1, Sex, S2.

Step	Intercept	Age	Sex	BMI	BP	S1	S2	S3	S4	S5	S6
1	-117	0.00	0.00	10.23	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	-299	0.00	0.00	7.28	0.00	0.00	0.00	0.00	0.00	56.06	0.00
3	-334	0.00	0.00	6.50	0.90	0.00	0.00	0.00	0.00	49.58	0.00
4	-327	0.00	0.00	6.53	0.93	-0.28	0.00	0.00	0.00	58.86	0.00
5	-320	0.00	-14.14	6.47	1.05	-0.30	0.00	0.00	0.00	60.36	0.00
6	-313	0.00	-21.59	5.71	1.13	-1.04	0.84	0.00	0.00	73.31	0.00

The process can be unstable, in that relatively small changes in the data might cause one variable to be selected instead of another, after which the sequence may be completely different. For example, Table 3 shows the result of the same procedure applied to a bootstrap sample of the original data (a sample of the same size, with replacement). As before, BMI and S5 are selected first, but now the sequence continues with Sex, S3, S6, and S4, rather than BP, S1, Sex and S2. Only three of the first six variables are common.

Table 3: Forward Stepwise Applied to a Bootstrap Sample: The sequence is BMI, S5, Sex, S3, S6, S4.

Step	Intercept	Age	Sex	BMI	BP	S1	S2	S3	S4	S5	S6
1	-117	0.00	0.00	10.23	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	-299	0.00	0.00	7.28	0.00	0.00	0.00	0.00	0.00	56.06	0.00
3	-293	0.00	-8.00	7.30	0.00	0.00	0.00	0.00	0.00	57.11	0.00
4	-194	0.00	-15.62	6.69	0.00	0.00	0.00	-0.91	0.00	51.48	0.00
5	-211	0.00	-17.08	6.42	0.00	0.00	0.00	-0.91	0.00	47.95	0.47
6	-203	0.00	-16.38	6.45	0.00	0.00	0.00	-1.20	-5.23	52.33	0.52

Backward stepwise regression is similar, except that initially all variables are included in the model, and variables are removed one at a time, according to which variable makes the smallest contribution to the fit.

Efroymson's procedure (Efroymson, 1960) combines forward and backward steps. Starting from some initial model, the worst included variable is discarded if the effect on fit falls short of some specified criteria; otherwise the best excluded variable is added if the improvement on fit exceeds a specified criteria.

The preceding algorithms are greedy, making the best change at each step, regardless of future effects. In contrast, all-subsets regression is exhaustive, considering all subsets of variables of each size. In practice the interest is only in some number of best subsets of each size, which are found efficiently by a leaps and bounds procedure (Furnival and Wilson, 1974) in the case of linear regression. The advantage over stepwise procedures is that the best set of two predictors need not include the predictor that was best in isolation. The disadvantage of leaps and bounds is that biases in inference are even greater, because it considers a much greater number of possible models.

In the case of linear regression, all computations for these stepwise and all-subsets procedures can be computed using a single pass through the data. This improves speed substantially in the usual case that $n >> p$, though not when $p > n$, as in the analysis of microarrays. Consider the model

$$Y = X\beta + \epsilon \tag{1}$$

where Y is a vector of length n, X an n by p matrix, β a vector of length p containing regression coefficients, and ϵ assumed to be a vector of independent normal noise terms. In variable selection, when some predictors are not included in a model, the corresponding terms in β are set to zero. There are a number of ways to compute regression coefficients and error sums of squares in both stepwise and all subsets regression. One possibility is to use the cross-product matrix

$$Z'Z = \left[\begin{array}{cc} X'X & X'Y \\ Y'X & Y'Y \end{array} \right], \tag{2}$$

where Z indicates the augmented matrix $[\ X \quad Y\ ]$. Another is to use the QR decomposition $Z = QR$, where Q has orthonormal columns and R is upper triangular. Both $Z'Z$ and QR with Q stored in a compact factored form can be computed in a single pass through the data, and in both cases there are efficient updating algorithms for adding or deleting variables (the SWEEP algorithm for $Z'Z$ and planar rotations for QR). Numerical accuracy in these computations can be assessed via the condition number (ratio of largest to smallest eigenvalue magnitudes) of the linear equations used to solve for the coefficients. The condition number of $Z'Z$ is of the order of the square of that for QR, so that QR is considerably more stable numerically. See e.g. (Kennedy and Gentle, 1980; Lange, 1999; Miller, 2002; Monahan, 2001; Thisted, 1988) for further information.

For nonlinear regressions, the computations are iterative, and it is not possible to fit all models in a single pass through the data.

Those points carry over to LARS. We summarize them briefly here, and return to them in more detail later. The original LARS algorithm computes $X'X$ and $X'Y$ in one pass through the data; using the QR factorization would be more stable, and could also be done in one pass, in the linear regression case. LARS for nonlinear regression requires multiple passes through the data for each step, hence speed becomes much more of an issue.

Ridge Regression

The ad-hoc nature of classical variable selection methods, together with observed instability in the associated regression coefficients and predictions, has led to other approaches.

One approach is ridge regression (Draper and Smith, 1998; Miller, 2002), which in its usual form includes all predictors, but with typically smaller coefficients than they would have under ordinary least squares. In the simplest form, the estimated regression parameters are equal to

$$\hat{\beta}_\theta = (X'X + \theta I)^{-1}X'Y, \tag{3}$$

where θ is positive scalar and I is a $p \times p$ identity matrix. With $\theta = 0$ this reduces to ordinary least-squares estimates (with all variables included). Otherwise it corresponds to minimizing a penalized sum of squares,

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{i,j}\beta_j)^2 + \theta \sum_{j=1}^{p} \beta_j^2. \tag{4}$$

In practice no penalty is applied to the intercept, and variables are scaled to variance 1 so that the penalty is invariant to the scale of the original data.

Figure 1 shows the coefficients for ridge regression graphically as a function of θ. As θ increases the terms become generally smaller. Those variables which are highly correlated with other variables are affected most, e.g. S1 and S2 have correlation 0.90.

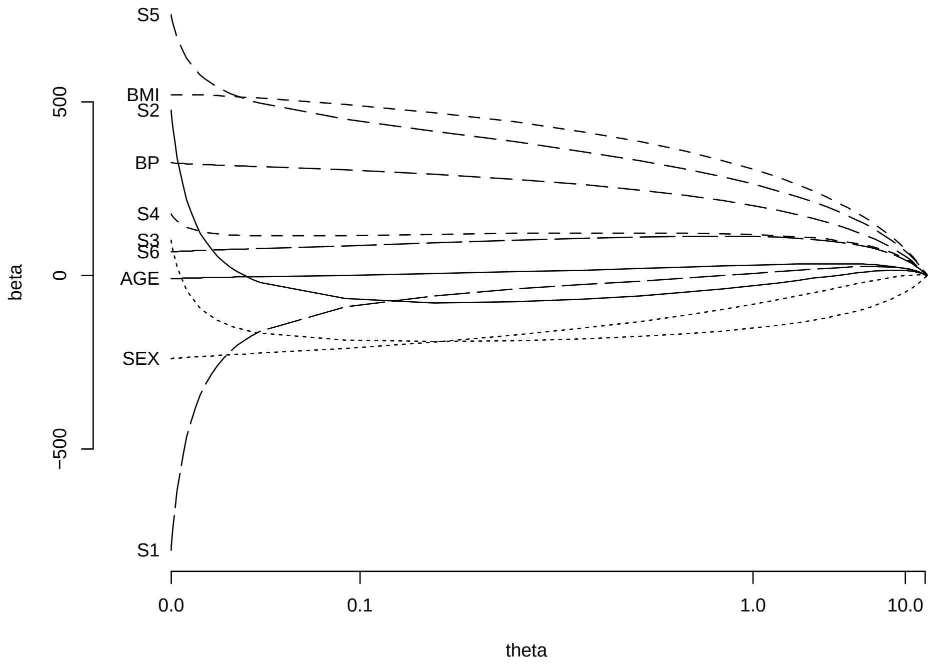

Figure 1: Coefficients for ridge regression, using standardized variables.

Note that as θ increases, the coefficients approach but do not equal zero. Hence, no variable is ever excluded from the model (aside from some cases where coefficients cross zero for smaller values of θ).

In contrast, the use of an L_1 penalty does reduce terms to zero. This yields the Lasso (Tibshirani, 1996), which we consider next.

Lasso

Tibshirani (Tibshirani, 1996) proposed minimizing the residual sum of squares, subject to a constraint on the sum of absolute values of the regression coefficients, $\sum_{j=1}^{p} |\beta_j| \leq t$. This is equivalent to an optimization problem with an L_1 penalty on the regression coefficients,

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{i,j}\beta_j)^2 + \theta \sum_{j=1}^{p} |\beta_j|. \tag{5}$$

Figure 2 shows the resulting coefficients. For comparison, the right panel shows the coefficients from ridge regression, plotted on the same scale. To the right, where the penalties are small, the two procedures give close to the same results. More interesting is what happens starting from the left, as all coefficients start at zero and penalties are relaxed. For ridge regression all coefficients immediately become nonzero. For the lasso, coefficients become nonzero one at a time. Hence the L_1 penalty results in variable selection, as variables with coefficients of zero are effectively omitted from the model.

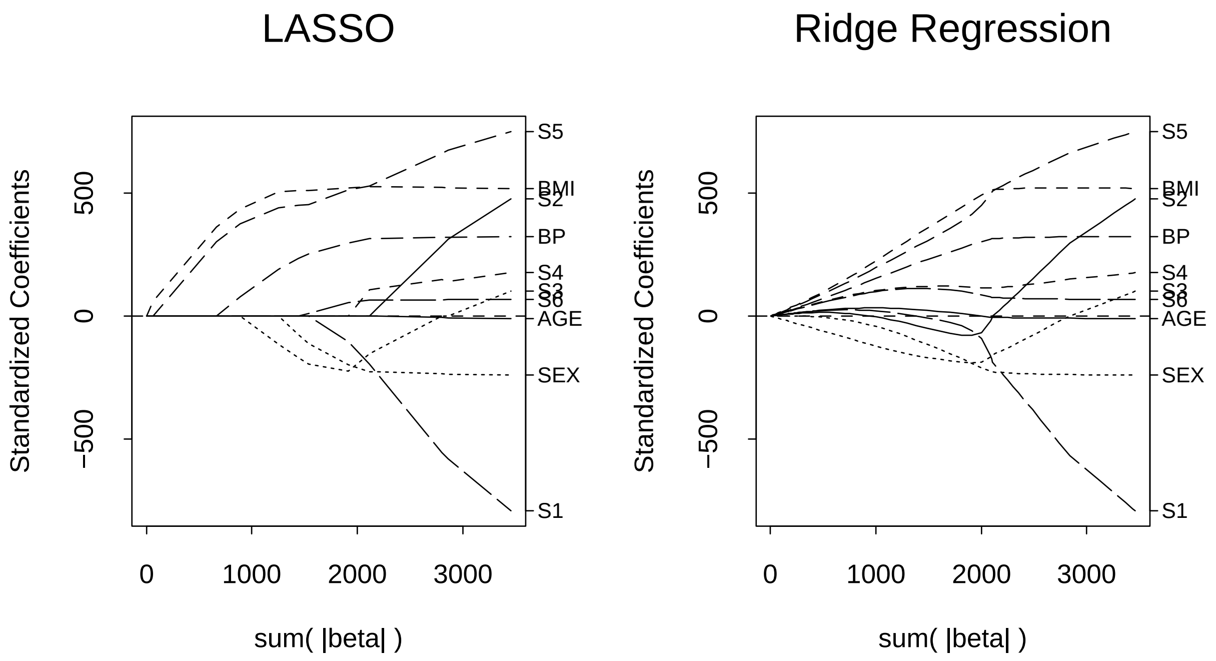

Figure 2: Coefficients for Lasso and Ridge Regression (L_1 and L_2 penalties).

Another important difference occurs for the predictors which are most significant. Whereas an L_2 penalty $\theta \sum \beta_j^2$ pushes β_j toward zero with a force proportional to the value of the coefficient, an L_1 penalty $\theta \sum |\beta_j|$ exerts the same force on all nonzero coefficients. Hence for variables which are most valuable, which clearly should be in the model and where shrinkage toward zero is less desirable, an L_1 penalty shrinks less.

While this second difference is not as important for interpreting models as the first, it may be even more important for providing accurate predictions of future values.

In this case, BMI (body mass index) and S5 (a blood serum measurement) appear to be most important, followed by BP (blood pressure), S3, Sex, S6, S1, S4, S2, and Age. Some curious features are apparent. S1 and S2 enter the model relatively late, but when they do their coefficients grow rapidly, in opposite directions. These two variables have strong positive correlation, so these terms largely cancel out, with little effect on predictions for the observed values. The collinearity between these two variables has a number of undesirable consequences—relatively small changes in the data can have strong effects on the coefficients, the coefficients are unstable, predictions for new data may be unstable, particularly if the new data do not follow the same relationship between S1 and S2 found in the training data, and the calculation of coefficients may be numerically inaccurate. A second curious feature is that variable S3 changes direction when S4 enters the model, ultimately changing sign. Here too, high (negative) correlation between S3 and S4 appears to be an explanation.

Forward Stagewise

We turn now to another procedure, forward stagewise regression, which at first glance appears to be very different from the Lasso, but turns out to have similar behavior.

This procedure is motivated by a desire to mitigate the negative effects of the greedy behavior of stepwise regression. In stepwise regression, the most useful predictor is added to the model at each step, and coefficients for the current predictors are estimated using ordinary unpenalized least-squares. Stepwise begins with all coefficients equal to zero, and predictors are retained in the model after they are added. Forward stagewise also begins with all coefficients equal to zero, and picks the same variable as stepwise at first. However, stagewise takes only a small step in the direction of that variable (increasing or decreasing the corresponding coefficient, according to whether the correlation between the variable and the current residuals is positive or negative). Stagewise then picks the variable with highest correlation with the current residuals (this may be the same variable as in the previous step), and takes a small step for that variable, and continues in this fashion.

Where one variable has a clear initial advantage over other variables there will be a number of steps taken for that variable. Subsequently, once a number of variables are in the model, the procedure tends to alternate between them.

When two variables are highly correlated and of nearly equal value as predictors, stagewise tends to alternate between the two variables, with each ending up with a coefficient half as large as it would have in isolation. Hence the forward stagewise algorithm is more stable than ordinary stepwise regression, for which one of the two coefficients would be zero and the other large.

While forward stagewise regression has a very different derivation than the Lasso, an idealized version (with the step size tending toward zero) has very similar behavior to the Lasso. In the diabetes example, it gives identical results until the eighth variable enters, after which there are small differences. Efron et al. (2004) discuss the differences.

There are also strong connections between forward stagewise regression and the boosting algorithm currently popular in machine learning (Efron et al., 2004; Hastie et al., 2001). Indeed, the difference is not really in the fitting method, but rather in the predictor variables used; in stagewise the predictors are typically determined in advance, while in boosting the next variable is determined on the fly. In each case the variables used as predictors in the model may be nonlinear functions of other variables, such as trees.

Least Angle Regression

Least angle regression (Efron et al., 2004) can be viewed as a version of stagewise that uses mathematical formulas to accelerate the computations. Rather than taking many tiny steps with the first variable, the appropriate number of steps are determined algebraically, until the second variable begins to enter the model. Then, rather than taking alternating steps between those two variables until a third variable enters the model, we jump right to the appropriate spot. Figure 3 shows this process in the case of 2 predictor variables, for linear regression.

The first variable chosen is the one which has the smallest angle between the variable and the response variable; in Figure 3 the angle COX_1 is smaller than COX_2. We proceed in that direction as long as the angle between that predictor and the vector of residuals $Y - \gamma X_1$ is smaller than the angle between other predictors and the residuals. Eventually the angle for another variable will equal this angle (once we reach point B in Figure 3), at which point we begin moving toward the direction of the least-squares fit based on both variables. In higher dimensions we will reach the point at which a third variable has an equal angle, and joins the model, etc.

Expressed another way, the (absolute value of the) correlation between the residuals and the first predictor

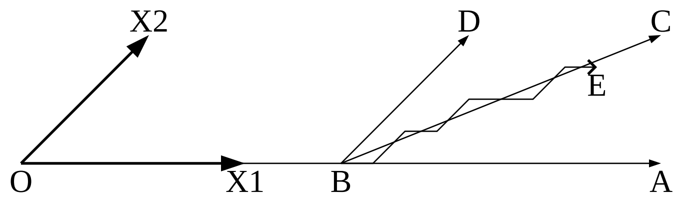

Figure 3: The LAR algorithm in the case of 2 predictors. O is the prediction based solely on an intercept. $C = \hat{Y} = \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$ is the ordinary least-squares fit, the projection of Y onto the subspace spanned by X_1 and X_2. A is the forward stepwise fit after one step; the second step proceeds to C. Stagewise takes a number of tiny steps from O to B, then takes steps alternating between the X_1 and X_2 directions, eventually reaching E; if allowed to continue it would reach C. LAR jumps from O to B in one step, where B is the point for which BC bisects the angle ABD. At the second step it jumps to C. The Lasso follows a path from O to B, then from B to C. Here LAR agrees with Lasso and stagewise (as the step size $\rightarrow 0$ for stagewise). In higher dimensions additional conditions are needed for exact agreement to hold.

is greater than the (absolute) correlation for other predictors. As γ increases, eventually another variable will have equal correlation with the residuals as the active variable, and joins the model as a second active variable. In higher dimensions additional variables will eventually join the model, when the correlation between all active variables and the residuals drops to the levels of the additional variables.

Three remarkable properties of LAR There are three remarkable things about LAR. First is the speed: Efron et al. (2004) note that "The entire sequence of LARS steps with $m < n$ variables requires $O(m^3 + nm^2)$ computations — the cost of a least squares fit on m variables."

Second is that the basic LAR algorithm, based on the geometry of angle bisection, can be used to efficiently fit the Lasso and stagewise models, with certain modifications in higher dimensions, (Efron et al., 2004). This provides a fast and relatively simple way to fit Lasso and stagewise models.

Madigan and Ridgeway (2004) comments that Lasso has had little impact on statistical practice, due to the inefficiency of the original Lasso and complexity of more recent algorithms (Osborne et al., 2000) and that this "efficient, simple algorithm for the Lasso as well as algorithms for stagewise regression and the new least angle regression" are "an important contribution to statistical computing".

Third is the availability of a simple C_p statistic for choosing the number of steps,

$$C_p = (1/\hat{\sigma}^2) \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 - n + 2k \tag{6}$$

where k is the number of steps and $\hat{\sigma}^2$ is the estimated residual variance (estimated from the saturated model, assuming that $n > p$). This is based on Theorem 3 in (Efron et al., 2004), which indicates that after k steps of LAR the degrees of freedom $\eta = \sum_{i=1}^{n} \text{cov}(\hat{\mu}_i, Y_i)$ is approximately k. Using this C_p statistic, one would stop after the number of steps k that minimizes the statistic.

Zou et al. (2004) extend that result to Lasso, showing an unbiased relationship between the number of terms in the model and degrees of freedom, and discuss C_p, AIC and BIC criterion for model selection.

There are some questions about this C_p statistic (Ishwaran, 2004; Loubes and Massart, 2004; Madigan and Ridgeway, 2004; Stine, 2004), and some suggest other selection criteria, especially cross-validation.

Note that there are different definitions of degrees of freedom, and the one used here is appropriate for C_p statistics, but that k does not measure other kinds of degrees of freedom. In particular, neither the average drop in residual squared error, nor the expected prediction error are linear in k (under the null hypothesis that $\beta_j = 0$ for all j) Figure 4 shows the behavior of those quantities. In the left panel we see that the residual sums of squares drop more quickly for LAR than for ordinary least squares (OLS) with fixed prediction order, suggesting that by one measure, the effective degrees of freedom is greater than k. In the right panel, the sums of squares of coefficients measures how much worse predictions are than using the using the true parameters $\beta_j = 0$; here LAR increases more slowly than for OLS, suggesting effective degrees of freedom less than k. These two effects balance out for the C_p statistic.

In contrast, stepwise regression has effective degrees of freedom greater than the number of steps; it overfits when there is no true signal, and prediction errors suffer.

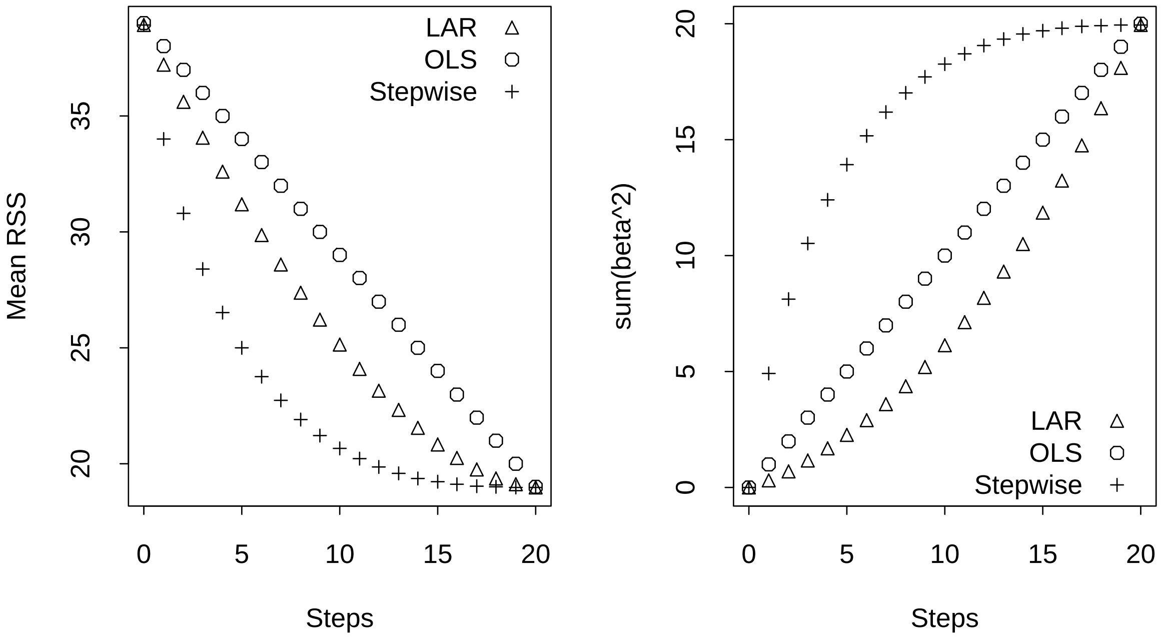

Figure 4: Effect of LAR steps on residual variance and prediction error. The left panel shows the residual sum of squares for LAR, ordinary least-squares with fixed predictor order, and stepwise regression. The right panel shows $\sum_{i=1}^{p} \beta_j^2$; this measures how much less accurate predictions are than for the true model. Figures based on simulation with 10,000 replications, with n = 40, p = 20, orthogonal predictors with norm 1, $\beta_j = 1 \forall j$, and residual variance 1.

These results are encouraging. It appears that LAR fits the data more closely than OLS, with a smaller penalty in prediction errors. While in this example there is only noise and no signal, it suggests that LAR may have relatively high sensitivity to signal and low sensitivity to noise. Work is needed to investigate this.

Comparing LAR, Lasso and Stagewise In general in higher dimensions native LAR and the least angle implementation of Lasso and stagewise give results that are similar but not identical. When they differ, LAR has a speed advantage, because LAR variables are added to the model, never removed. Hence it will reach the full least-squares solution, using all variables, in p steps. For Lasso, and to a greater extent for stagewise, variables can leave the model, and possibly re-enter later, multiple times. Hence they may take more than p steps to reach the full model. Efron et al. (2004) test the three procedures for the diabetes data using a quadratic model, consisting of the 10 main effects, 45 two-way interactions, and 9 squares (excluding the binary variable Sex). LAR takes 64 steps to reach the full model, the Lasso variation takes 103, and stagewise takes 255. Even in other situations, when stopping short of the saturated model, LAR has a speed advantage.

The three methods have interesting derivations. Lasso is regression with an L_1 penalty, a relatively simple concept; this is also known as a form of regularization in the machine learning community. Stagewise is closely related to boosting, or "slow learning" in machine learning. LAR has a simpler interpretation than the original derivation; it can be viewed as in relation to Newton's method, which makes it easier to extend to some nonlinear models such as generalized linear models.

3.2.1 Existing knowledge

We begin with a review of other contributions in the literature, followed by a summary of work needed.

Other penalty approaches Ridge regression uses an L_2 penalty, and Lasso an L_1 penalty. Zou and Hastie (2005a) propose the "elastic net", penalized regression with a sum of L_1 and L_2 penalties. This is useful in the analysis of microarray data, as it tends to bring related genes into the model as a group. It appears to give better predictions than Lasso when predictors are correlated.

Tibshirani et al. (2005) propose the "fused Lasso", involving a combination of an L_1 penalty on coefficients, and an L_1 penalty on the difference between adjacent coefficients. This is useful for problems such as the analysis of proteomics data, where there is a natural ordering of the predictors (e.g. measurements on different wavelengths) and coefficients for nearby predictors should normally be similar; it tends to give locally-constant coefficients.

Yuan and Lin (2006) discuss "grouped Lasso" and "grouped LARS", for use when some predictors have multiple degrees of freedom, such as factor variables. The approach is similar to the approach suggested in our Phase I application and implemented in our Phase I work for LAR; involving working with the angle between the partial residuals and the space spanned by a basis for the a factor, rescaled based on the degrees of freedom.

Yuan and Lin (2005) propose an empirical Bayes method closely related to Lasso. Unlike Efron et al. (2004), where the error variance needed for the C_p statistic is estimated from the saturated model, here it is estimated with other parameters, so the method could be used when $p > n$.

Nonlinear models The original LARS method is for linear regression. Several authors have discussed extensions to other models, including Cox regression (Gui and Li, 2005; Park and Hastie, 2006a), generalized linear models (Madigan and Ridgeway, 2004; Park and Hastie, 2006a), robust linear regression (Rosset and Zhu, 2004a; Van Aelst et al., 2005), exponential family models (Rosset, 2005), and support vector machines (Hastie et al., 2004; Zhu et al., 2003).

Some additional authors discuss general strategies for solutions in nonlinear models. Roth (2004) discusses a method for iteratively reweighted least squares (IRLS) applications. Rosset and Zhu (2004b) discuss conditions under which coefficient paths are piecewise linear, and Rosset (2005) discuss a method for tracking curved coefficient paths; however, the algorithm requires computing a gradient and Hessian at each of many small steps, and so is poorly suited for large problems. Kim et al. (2005a) propose a gradient approach particularly useful for high dimensions.

Applications A number of authors discuss applications to microrarrays (Gui and Li, 2005; Segal et al., 2003; Zhu and Hastie, 2004; Zou and Hastie, 2005a), or proteomics (Tibshirani et al., 2005).

Work needed LARS has considerable promise, offering speed, interpretability, relatively stable predictions, close to unbiased inferences, and nice graphical presentation of the whole sequence of coefficients.

But considerable work is required to turn this promise into widely-used reality.

- A number of different algorithms have been developed, for linear and nonlinear models. These differ in speed, numerical stability, accuracy (in the nonlinear case, how well do algorithms track the exact curved coefficient paths), collinearity, and handling of details such as variables which are nearly tied in importance. Additional algorithms are undoubtedly being developed. Work is needed to compare the algorithms, with artificial and real data, with a variety of sizes — large and small n and p.

  Speed is an issue for nonlinear models, particularly if cross validation is used for model selection, or bootstrapping for inferences. In the linear regression case the cross-product matrices or QR decomposition required for computations can be calculated in a single pass through the data. In contrast, for the nonlinear models, fitting each subset of predictors requires multiple passes through the data, with the QR or cross-product summary recomputed at each pass.
- Alternate penalties such as the elastic net and fused lasso offer advantages for certain kinds of data, in particular microarrays and proteomics; work is needed to create algorithms using these penalties in nonlinear models, to investigate their properties, and to provide guidance on choosing the tuning parameters—in contrast to LAR and Lasso, which each have only a single tuning parameter, these procedures have two or more.
- The original methodology is limited to continuous or binary covariates. The grouped Lasso and LAR are one extension to factor variables or other variables with multiple degrees of freedom such as polynomial and spline fits. Work is needed to investigate these methods, and to extend them to nonlinear models.
- There are a number of practical considerations in some applications that need attention, including order restrictions (e.g. main effects should be included in a model before interactions, or linear terms before quadratic), forcing certain terms into the model, allowing unpenalized terms, or applying different levels of penalties to different predictors based on an analyst's knowledge. For example, when estimating a treatment effect, the treatment term should be forced into the model and estimated without penalty, while covariates should be optional and penalized.
- A variety of work is needed under the broad category of inferences, including tuning parameters and more traditional inferences. LARS and Lasso require the choice of a tuning parameter (the number of steps, or magnitude of the L_1 penalty); the elastic net, and fused Lasso require multiple tuning parameters. Work is needed to investigate and compare methods including C_p, AIC, BIC, cross-validation, and empirical Bayes. The theoretical work on the C_p statistic to date is under the null hypothesis that no coefficients are nonzero; how is it affected when some coefficients are nonzero?
- Work is needed to develop estimates of bias, standard error, and confidence intervals, for predictions, coefficients, and linear combinations of coefficients. Are predictions sufficiently close to normally-distributed to allow for the use of t confidence intervals? Coefficients are definitely not normally distributed, due to a point mass at zero; but when coefficients are sufficiently large, might t intervals still be useful?
- Work is also needed to look at the signal-to-noise ratio for these methods, and to compare to alternatives. A good signal-to-noise ratio would be a strong impetus for the statistical community to use the methods.
- Work is needed to develop numerical and graphical diagnostics to interpret regression model output.
- Finally, to truly realize the promise of these methods, they must be encoded in robust and easy-to-use software suitable for a broad base of users, not just sophisticated academic researchers.

We close this section by returning to a positive note, with comments in the literature about LARS: Knight (2004) is impressed by the robustness of the Lasso to small changes in its tuning parameter, relative to more classical stepwise subset selection methods, and notes "What seems to make the Lasso special is (i) its ability to produce exact 0 estimates and (ii) the 'fact' that its bias seems to be more controllable than it is for other methods (e.g., ridge regression, which naturally overshrinks large effects) . . . " Loubes and Massart (2004) indicate "It seems to us that it solves practical questions of crucial interest and raises very interesting theoretical questions ...". Segal et al. (2003) write "The development of least angle regression (LARS) (Efron et al., 2004) which can readily be specialized to provide all lasso solutions in a highly efficient fashion, represents a major breakthrough. LARS is a less greedy version of standard forward selection schemes. The simple yet elegant manner in which LARS can be adapted to yield lasso estimates as well as detailed description of properties of procedures, degrees of freedom, and attendant algorithms are provided by (Efron et al., 2004)."

3.2.2 Commercial Opportunities

(Some text is omitted from this version of the proposal.)

3.2.3 Societal Benefits

This project could have a strong positive impact on our nation's health care. Predicting outcomes based on covariates, and determining which covariates most affect outcomes, are among the most important problems in biostatistics.

This proposed software should enable medical researchers to obtain stable and interpretable models with better prediction accuracy in high-dimensional situations, including analysis of data from the emerging technologies such as microarrays and proteomics.

3.3 Phase II Applications and Anticipated Outcomes

The overall goal in Phase II is to produce high-quality software for classification and regression based on LARS for widespread use. This project should substantially widen the rather limited scope of the original software. We intend to handle many of the issues important in practice, including missing data, assessing significance and stability.

4 Preliminary Studies/Progress Report

4.1 Phase I Grant Performance Period

(Some text is omitted from this version of the proposal.)

4.2 Key Personnel and Project Service

(Some text is omitted from this version of the proposal.)

4.3 Specific Aims

The three specific aims in the Phase I proposal are:

- extension to logistic regression
- allow factor variables with more than two levels
- develop efficient and numerically stable computation

These aims were chosen to demonstrate feasibility of extending the basic methodology to produce software that could be widely used in practice for a variety of applications. The extension to logistic regression provides evidence that it is feasible to extend the basic methodology to other nonlinear regression models, including other generalized linear models, survival analysis, and mixed-effects models. Factor variables are important in their own right, and the extension to factor variables suggests that the methodology can be extended to other types of predictors with multiple degrees of freedom, such as nonlinear relationships using polynomials or splines. It is also the first step in extending the methods to hierarchical models, for example to fit main effects before interactions. Finally, the development of numerically-stable algorithms is necessary for reliable use by a wide range of analysts.

4.4 Results, Progress, and Achievement Towards Goals

We achieved the three specific aims, and substantially more, both methodological developments and software. We accomplished more than anticipated in terms of handling collinearity, simultaneous entry of variables, different ways of handling factors, and developing an alternate formulation of LARS that supports nonparametric regression such as generalized additive models. Furthermore, the software produced is substantially more complete than anticipated.

Here we discuss two issues, give an overview of the main results, then give more detail in key areas.

The first issue is that we decided to focus our efforts on the LAR and LASSO cases, and de-emphasize the Forward Stagewise case. LAR is the fastest, and LASSO has a nice interpretation in terms of optimizing a penalized likelihood. Forward stagewise is the slowest, sometimes substantially so, and does not seem to have strong advantages.

Second, we decided to produce an open-source library that will run in both S-PLUS and R. This makes it easier to benefit from open-source work done in the academic community, and improves our ability to work collaboratively with outside contributors; indeed, the decision was greeted enthusiastically by key potential collaborators, and a large number of researchers in the area have requested the prototype library.

Now, we give an overview of our main results. We began with the basic algorithm of (Efron et al., 2004) and their publicly-available software (Efron and Hastie, 2003), for the linear regression case. We created a version that is substantially faster ($\sim$16 times faster), primarily by doing core calculations in FORTRAN rather than S (we use S to refer to code in the S language, of which S-PLUS and R are dialects).

We developed a more accurate algorithm for the linear case, using the QR decomposition of X rather than the Cholesky decomposition of cross-product matrices $X'X$. For example, with $n = 100$, $p = 50$, and correlation 0.999999 between each pair of predictors, the new algorithm has mean absolute roundoff error of 7.5×10^{-13}, compared to 1.8×10^{-8} for the original algorithm. The ratio between these numbers is 24346, approximately equal to the condition number of X, 21760, as expected. Note that in practice it is easy to get even higher condition numbers even with lower pairwise correlations, due to higher-dimensional collinearity.

The more accurate algorithm is slower than the original ($\sim$4.8 times slower), in the current S implementation. Converting this to FORTRAN or some other compiled language should result in a substantial speedup, even more than the factor of 16 observed for the other algorithm, because we could use QR updating techniques when adding or dropping variables, whereas the prototype S implementation recomputes the decomposition from scratch in each step. We expect the ultimate speed to nearly equal the original algorithm.

We also made advances in an area related to accuracy, but not anticipated in the Phase I proposal — linear dependence. When predictors are linearly dependent then an ordinary linear least squares fit is ill-defined, unless predictors are dropped from the model. Similar, in LARS, the set of active variables should not be linearly dependent. Our software checks for linear dependence, and excludes dependent inactive variables from consideration for activation, permanently in the LAR case, and temporarily in the LASSO case (when a variable leaves a model, the set of variables that should be prohibited may change).

For logistic regression, we developed an algorithm for the LAR case. We also began work on a version for the LASSO, but when the `glmpath` library (Park and Hastie, 2006a) was released, we decided to use that since it included more sophisticated computations for the underlying convex optimization problem than we would have had time to develop in Phase I.

For factor variables, we took two approaches. The first, suggested by Saharon Rosset, is to include a dummy variable for each level of a factor, and let the fitting algorithm select coefficients; only some of the levels would have nonzero coefficients. The second is more closely related to the approach envisioned in our Phase I proposal, incorporating a factor variable as a single entity.

We created a software library including documentation that runs in both S-PLUS and R. The main fitting routines in the library are:

- `lars.fit.eh` the original Efron-Hastie algorithm, in S; LAR, LASSO and forward stagewise,
- `lars.fit.fortran` FORTRAN version of the original algorithm, LAR, LASSO and forward stagewise,
- `lars.fit.s` more accurate algorithm, in S; LAR and LASSO,
- `glars.fit.s` logistic regression, in S; LAR,
- `glmpath` logistic, linear, and Poisson regression, calls FORTRAN for core calculations; LASSO, and
- `coxpath` Cox proportional hazards regression, calls FORTRAN for core calculations; LASSO.

The `lars` function provides a user-friendly front end to the three fitting routines for the linear case. It allows the user to specify variables to use by means of a formula, rather than constructing a design matrix manually. This function supports factor variables using the dummy variable approach (the second approach to factors currently requires calling `lars.fit.s` directly).

There are also some routines for plotting and nicely-formatted output of the fitting results, or further analysis such as cross-validation.

The `lars.fit.eh` function is from (Efron and Hastie, 2003); the `glmpath` and `coxpath` functions are from (Park and Hastie, 2006a). The plotting, printing, and cross-validation routines are also largely from those libraries. We have made some improvements, ranging from the obvious (allow space for axis labels so they are not off the page) to more subtle (avoiding programming constructions that fail for some user inputs).

Extension to logistic regression. Our extension of LARS to logistic regression exploits the relationship of LARS to Newton's method, which also leads to efficient and numerically stable computational procedures in the linear case.

As an example of the effectiveness of this approach, our test cases included the 2-class leukemia data (Golub et al., 1999), which consists of training and test sets with 38 and 34 observations, respectively, and 3051 genes as predictors. Collinearity is an issue here, and our implementation selected a maximum of only 8 out of the 3051 genes. A summary of results for the best models obtained is given in Table 4.

Table 4: Classification errors for the logistic LAR fit to the training data on the 2-class leukemia data.

# predictors (genes)	# training errors	# test errors
6 / 3051	0 / 38	1 / 34
7 / 3051	0 / 38	1 / 34
8 / 3051	0 / 38	2 / 34

The problem to be solved is to fit to a vector y with a linear or generalized linear model

$$y \sim \text{model}(X\beta); \quad X \equiv \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix}; \quad \beta \equiv \begin{pmatrix} \beta_1, & \dots, & \beta_p \end{pmatrix}^T$$

in such a way that the set of coefficients $\beta_1, \dots, \beta_p$ contains relatively few nonzero components. In the case of linear LARS, the model is chosen is with respect to the sum of squares criterion

$$F_{\text{least-squares}}(\beta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 \tag{7}$$

in increments of only a few variables at a time while for logistic LARS the model is chosen with respect to the following criterion

$$F_{\text{logistic}}(\beta) = \sum_{i=1}^{n}\left[\log\left(1+\exp(x_i^T\beta)\right) - y_i\left(x_i^T\beta\right)\right], \tag{8}$$

(e.g. McCullagh and Nelder 1989, p. 115). For ordinary least-squares and logistic regression, coefficients are chosen to minimize these criteria; however LARS generates a series of model fits that are related to these criteria, but do not in general produce the same coefficients as least-squares or logistic fits for the model reduced to the set of predictors with nonzero coefficients. The advantage in viewing LARS in relation to Newton's method is that the derivations extend in a straightforward manner to other generalized linear models of interest.

Efficient and numerically stable computation. .

Computation of LARS directions. As mentioned above, LARS directions are related to the Newton directions relative to the active predictors with respect to the objective function (7) in the linear case and (8) in the logistic case. The Newton direction d is the solution to

$$\nabla^2 F(\beta)\, d = -\nabla F(\beta),$$

where the objective, Hessian and gradients are given in Table 5, in which X denotes the current but not necessarily complete set of predictors. We assume that any new predictors are added as the last columns of X and that the coefficients in β corresponding to the new columns are initially 0. Two mathematically equivalent forms are given in the logistic case; the choice is important for numerical reasons, as explained later in this section.

Table 5: Linear and logistic LARS, objective, gradient and Hessian.

	t_i : objective $= \sum_{i=1}^n t_i$	r_i : gradient $= -X^T r$	s_i : Hessian $= X^T \mathrm{diag}(s) X$
linear	$\frac{1}{2}(y_i - x_i^T \beta)^2$	$y_i - x_i^T \beta$	1
logistic	$\log\left[1+\exp(x_i^T\beta)\right] - y_i\left(x_i^T\beta\right)$	$y_i - \frac{\exp(x_i^T\beta)}{1+\exp(x_i^T\beta)}$	$\frac{\exp(x_i^T\beta)}{\left[1+\exp(x_i^T\beta)\right]^2}$
logistic	$\log\left[1+\exp(-x_i^T\beta)\right] - \log\left[\exp(-x_i^T\beta)\right] - y_i\left(x_i^T\beta\right)$	$y_i - \frac{1}{1+\exp(-x_i^T\beta)}$	$\frac{\exp(-x_i^T\beta)}{\left[1+\exp(-x_i^T\beta)\right]^2}$

In the case of *linear LARS*, the Newton equations are

$$X^T X d = X^T (y - X\beta) = X^T r(\beta).$$

Note that the components of the negative gradient, $X^T r(\beta)$, when scaled by the norms of the rows of X, are the correlations for the current predictors. A unit step along the Newton direction would give a new set of coefficients $\beta + d$ for which the correlations corresponding to the predictors in X vanish. However in LARS, we move along the Newton direction only until the LAR, LASSO or forward stagewise criterion are met, after which we proceed with respect to a new set of predictors.

Numerically stable computation and efficient updating of the Newton direction is possible via the QR factorization of X, in which X is expressed as the product of an orthogonal matrix Q and an upper triangular matrix R. Once X is factored, the Newton direction can be efficiently computed by forming $u = Q^T r(\beta)$ then solving $Rd = (u_1, \ldots, u_p)^T$ (see, e.g. (Golub and Loan, 1996)).

Besides efficient solution of the Newton equations, the QR factorization has other advantages:

- numerical stability;
- allows detection of collinearity through small diagonal elements in R;
- can be updated efficiently to add or delete columns (predictors).

In the case of *logistic LARS*, the Newton equations are

$$X^T \mathrm{diag}(s(\beta)) X d = X^T D(\beta) X = X^T r(\beta), \tag{9}$$

where r and s are given in Table 5. Because of the nonlinearity, optimization with respect to the current set of predictors is iterative, and can be accomplished by IRLS. Assuming none of the diagonal elements of $D(\beta)$ vanish, the Newton equations (9) can be written as a linear least-squares problem

$$X^T D(\beta) X = X^T D(\beta)^{1/2} D(\beta)^{-1/2} r(\beta),$$

which can be solved via the QR factorization of $D(\beta)^{1/2}X$, and as described above for the linear case with $r(\beta)$ replaced by $D(\beta)^{-1/2}r(\beta)$. When $D(\beta)$ has small elements, we still want to form the QR factorization of $D(\beta)^{1/2}X$ because it is a stable way to obtain the Cholesky factor of $X^T D(\beta) X$, which is the just the upper triangular matrix R of the QR factorization. We compute the Newton direction d by first solving $R^T u = X^T r(\beta)$, then solving $Rd = u$. Small elements in $D(\beta)$ are then either indicative of a close fit to a 0 or 1 response, while small elements in the diagonal of R of are indicative of collinearity in the predictors, which may not be present at the start of the iteration but may arise in the process due to nonlinearity. In IRLS for ordinary logistic regression, the Newton directions d and coefficients β are repeatedly updated $\beta \leftarrow \beta + d$ until the gradient vanishes. In LAR we compute the ordinary coefficients, and take a LAR step in the direction $\Delta\beta$ between the previous coefficients and the ordinary coefficients (this difference is the sum of the Newton/IRLS steps). Other methods are are possible, e.g. the predictor-corrector approach of (Park and Hastie, 2006a).

Computing LARS step lengths. After computing the LARS direction, a step length along that direction is determined yielding either the LAR, LASSO, or stagewise coefficients. For brevity, we describe only the LAR case in this section; for LASSO or stagewise the step is further restricted.

Let $\mathcal{P}$ represent current set of predictors. A unit step length along the Newton direction (linear case) or along the sum of the IRLS directions (logistic case) would yield coefficients for which correlations (and gradient) with respect to $\mathcal{P}$ vanishes. In LAR, we take the smallest step length along this direction until a correlation in $\mathcal{P}^c$ is equal in magnitude to at least one correlation in $\mathcal{P}$, but remains smaller than or equal to all of the other correlations in $\mathcal{P}$. Since $\mathcal{P}$ includes those predictors with correlations that are the largest in magnitude after the previous step, and all correlations in $\mathcal{P}$ approach zero along $\Delta\beta$, this condition is guaranteed to be met.

It is important to include as new predictors all correlations that are close in magnitude to those of the current predictors; otherwise the method will take small steps (we have observed this in the original algorithm) and could possibly cycle due to roundoff error.

Gradient and Hessian computation for logistic LARS. For our logistic regression computations, we need the gradient and Hessian of the criterion (8), which can be expressed in terms of either $\exp(x_i^T\beta)$ or $\exp(-x_i^T\beta)$ and is given in Table 5. Because of the exponential, there is the potential for overflow and/or underflow[1] in these computations. For stability, we use the following strategy:

	$r_i \mid \nabla F_{\text{logistic}} \equiv -X^T r$	$s_i \mid \nabla^2 F_{\text{logistic}} = -X^T \text{diag}(s) X$
$x_i^T b \in [0, -\log(fl_{\text{MIN}})]$	$y_i - \frac{1}{1+\exp(-x_i^T\beta)}$	$\frac{\exp(-x_i^T\beta)}{\left[1+\exp(-x_i^T\beta)\right]^2}$
$x_i^T b \in [-\log(fl_{\text{MAX}}), 0]$	$y_i - \frac{\exp(x_i^T\beta)}{1+\exp(x_i^T\beta)}$	$\frac{\exp(x_i^T\beta)}{\left[1+\exp(x_i^T\beta)\right]^2}$
$x_i^T b > -\log(fl_{\text{MIN}})$	$y_i - 1$	0
$x_i^T b < -\log(fl_{\text{MAX}})$	$y_i - 0$	0

where fl_{MIN} is the smallest positive nonzero floating-point number, and fl_{MAX} is the largest positive floating point number for which fl_{MAX} and $-fl_{\text{MAX}}$ are both representable, respectively.[2] The extreme cases (which would not necessarily be rare because they correspond to close fits), can cause the Hessian to be singular or nearly singular, in which case the numerically stable procedures described in this section come into play.

[1]While occasional underflow is not usually problematic in computation, our computations are iterative procedures in which it is possible that very large numbers of underflows could occur, causing the hardware to trigger an error condition.

[2]Machines in common use are IEEE compliant, so that to four decimal places $\log(fl_{\text{MAX}}) = 709.7827$ and $\log(fl_{\text{MIN}}) = -708.3964$.

Factor Variables. We took two approaches to handling factor variables; the first splits a factor into dummy variables, the second keeps it as a group.

The first was suggested by our consultant Saharon Rosset:

> About the factor issue: why not penalize all coefficients (for all levels of factor)? I don't think any contrast should be used—just include indicator variables for all levels of the factor (linear dependence is not a problem, of course). Given the sparseness property of L_1 regularization I think that trying to add the whole factor as one variable would make no sense—let the penalty select the relevant factor levels and add them to the model.

This is easy to implement, provided care is taken to handle collinear variables. It can also lead to more interpretable models in some situations; e.g. when the resulting model has only a few non-zero coefficients for a factor with many levels, that suggests to the analyst that the levels with nonzero coefficients stand out.

However, possible non-uniqueness of the solution could make such interpretation misleading. For example, with two levels, either level could be picked to have a nonzero coefficient, and the resulting predictions and penalized objective function would be the same; for that matter, a linear combination of the two vectors of coefficients would also yield the same predictions and objective. For $p = 2$ there is never a unique solution, while for $p = 3$ the solution is always unique. For $p > 3$, there is a unique solution in some case, but not in others. A possible remedy would be the addition of a small L_2 penalty to make solutions unique (this would be an elastic net).

The second approach is to treat a factor as a whole. Instead of the angle between residuals and a predictor, we work with the angle between residuals and the projection of the residuals onto a η-dimensional space spanned by column vectors of X_j, where X_j is a design matrix for the factor, with η degrees of freedom (e.g. $\eta = 3$ for a factor variable with four levels). In LAR, rather than working with correlation when deciding which variable to add or in determining step length, we divide the correlation by $\sqrt{\eta}$, to provide a penalty for the extra degrees of freedom.

In the linear regression case, we find that the coefficient paths are linear, but the step length sometimes requires solving a nonlinear equation.

In the Lasso case, we propose to generalize the Lasso penalty $\theta \sum_{j=1}^{p} |\beta_j|$ as in (5), to

$$\theta \sum_{j=1}^{p} \sqrt{\eta_j} \|z_j\|, \tag{10}$$

where z_j is the contribution of variable j to the model and $\|z_j\|$ its L_2 norm, after subtracting its mean. This is equivalent for an ordinary predictor that is centered and scaled to norm 1, with $z_j = \beta x_j$. For a factor, $z_j = X_j \beta_j$, where β_j is a vector of coefficients of length η.

After implementing the LAR case and developing the Lasso proposal, we discovered that Yuan and Lin (2006) proposed the group LAR and group Lasso; the former is equivalent to our method and the later would give equivalent results, albeit with a different formulation. They note that the coefficient paths in group Lasso are not piecewise linear.

Penalty based on partial predictions. Our formulation (10), which involves a penalty based on the magnitude of partial predictions rather than coefficients, is interesting in its own right.

First, it makes it clear that one need not rescale predictors to have mean 0 and norm 1 prior to fitting the model. (For variables that are not centered, the penalty would be be $z_j - \overline{z}_j$, where $\overline{z}_j$ is the mean of vector z_j.) This has computation advantages, because rescaling takes time and memory, particularly in

IRLS, where every iteration would require rescaling with different weights. We anticipate taking advantage of this in Phase II.

Second, this provides a way to extend LARS to include certain nonparametric methods. For example generalized additive models (Hastie and Tibshirani, 1990), fit models of the form $\hat{y} = \sum_{i=1}^{j} f_j(x_j)$ where the f_j are smooth functions, often selected by smoothing splines or other smoothers. (10) generalizes immediately to $\theta \sum_{j=1}^{p} \sqrt{\eta_j} \| f_j(x_j) \|$, with η_j set to the equivalent degrees of freedom for the smoother.

5 Research Design and Methods

Our specific aims and preliminary timetable are listed in Section 2. Here we give additional information on our plans.

First phase For the first phase, our plans are predicated on the fact that this is a rapidly developing field, with the possibility of substantial outside collaboration from academics. Hence our efforts will be focused on creating an attractive platform for collaborative work. There are two parts to this.

The first part is the creation of an extendable library. Our goal is to create a framework that is attractive for outside collaborators to work in and extend. This framework should include appropriate front-end functions, e.g. `lars` for linear regression, `glars` for generalized linear models, and `coxlars` for proportional hazards regression. These front ends should handle initial data massaging (subsetting, exclusion of missing values if that option is chosen), selection of variables according to a user-specified formula, processing of factor variables, polynomials, spline terms, interactions, etc., then call a fitting routine. In contrast to the existing academic software, where functions are organized primarily for the convenience of the developer, the front-end functions should mimic the user interface of functions analysts are used to, such as `lm`, `glm`, and `coxph`, for linear, generalized linear, and cox regression, respectively.

The fitting routines are one opportunity for outside collaboration; they may be written by anyone, provided they follow certain guidelines (to be developed) for input and output. In the prototype library, one of the fitting routines is `lars.fit.eh`, the original LARS algorithm as coded by Efron and Hastie (Efron and Hastie, 2003).

The fitting routines should accept certain common inputs, e.g. `x` and `y`. Argument `type` indicates the type of fit to be performed, e.g. "lasso", "lar", "forward.stagewise", "elastic.net", "fused.lasso" (a routine need not support all of these; it could support only one method, possibly of the author's own invention). A fitting routine may have other inputs, specific to the routine.

The fitting routines should produce an object with certain components including a matrix of coefficients (one row for each step, and one column for each variable), vector of C_p values, vector of goodness of fit statistics (e.g. residual sum of squares for the linear case), etc. They may have additional components.

The object must also have a particular *class*, e.g. "lars" for the linear case, and possibility additional classes of the author's invention. The purpose of these classes relates to the overall design of S. S is an object-oriented system, and every object has a class. For example, objects created by the `glm` function (generalized linear models) have class "gam". A user typically calls the `glm` function and saves the results in an object, say "fit", then proceeds with additional analysis—printing the coefficients, performing diagnostic plots, comparing the fit to fits with additional variables, etc. These operations are performed by calling additional functions, e.g. `plot(fit)`. `plot` is a *generic function*; it takes advantage of the class of the object in deciding what operations to perform. Usually it does so by calling a *method* specifically designed for that class. In this case, `plot` calls method `plot.glm`, which performs appropriate plots for a "glm" object.

The library should have methods for "lars" objects for common generic functions such as `plot`, `print`, `coef` (to extract the coefficients), and others. Hence collaborators who write fitting methods that produce "lars" objects, and have the right components for a "lars" object, need not duplicate our effort by writing all these functions themselves.

On the other hand, the methods provide an additional opportunity for outside collaborators. They may improve our methods, create methods for additional generic functions, or create methods for new classes—if the object created by their fitting function has special features, they may give the object a class in addition to "lars", and write methods specifically for the class.

To summarize, we plan to create a library with some front-end functions to handle user input, some example fitting routines, and some common back-end functions for plotting or other operations. Collaborators may create their own fitting routines that follow our specifications, and may also contribute to the back-end functions.

(Some text is omitted from this version of the proposal.)

The second part to our initial work is creating fitting routines. This involves refining our current prototype routines (e.g. rewriting the accurate algorithm in FORTRAN), and ensuring that they meet our specifications, and adding key additional routines, such as LAR for Cox regression. This will be done primarily by Fraley and the scientific programmer.

A major goal in the initial effort is to design a library that collaborators can add to without extensive work on our part. Our experience in working with academic collaborators is that their code may be "90% complete", but that the remaining 10% of the work actually takes 90% of the overall time. We aim to set up guidelines so that they can make contributions that require no or minimal effort on our part to complete. The fitting routines we create should serve as examples.

The initial library should be in the form of a package that can be run from both S-PLUS and R. In order to allow collaborators who currently use R to ensure that their contributions will run in S-PLUS, Insightful will provide free time-limited S-PLUS academic licenses for collaborators, chosen by the PI.

Second phase A major goal of the second period is to begin to democratize the project — to move beyond academic collaborators to broader use. Toward that end, we anticipate creating a preliminary manual, and adding a graphical user interface (Windows S-PLUS version only), as well as refining the earlier library based on feedback.

This period also marks the beginning of broader outreach. In the first period our outreach would focus on collaborators; in this period we would reach out to additional groups, beginning with Insightful consultants and pre-sales engineers, and giving conference talks.

We anticipate expanding our current website for this project, `http://www.insightful.com/Hesterberg/glars`, to include links to documents that give an introduction to the area, citations and abstract, links to web sites of collaborators and others doing related work, links to related S-PLUS and R software, and other items to promote the project and the methodology. A list of open topics would be interesting to graduate students and researchers, and provide a way to direct research in directions we feel are important.

On the scientific side, we anticipate adding support for factor, polynomials, and splines to at least some of the models, adding additional models, and probably improving our earlier models.

(Some text is omitted from this version of the proposal.)

Third phase A major theme for this period is to begin to shift from methodology to applications.

One step is creating an interface between S+GLARS and S+ARRAYANALYZER and BIOCONDUCTOR, that would support important data structures common between both S+ARRAYANALYZER and BIOCONDUC-

TOR including `eset` (basic Affy data storage) and `marray` (for cDNA or long oligo arrays). Example data sets found in both S+ARRAYANALYZER and BIOCONDUCTOR could then be used to create case studies of microarray data. other Insightful products—details omitted.

Another step is creating a large-data version of the core algorithms, and creating an interface to the `bigdata` library in S-PLUS. This would be built on the large-data QR algorithm in the `bigdata` library.

This period should also mark the beginning of creation of training materials, incorporating the case studies.

Important methodological extensions include elastic net capability (for microarray data) and fused lasso (for proteomic data). Additional extensions may include support for handling missing data using multiple imputations (using the S+missing library), certain mixed-effects models, robust regression, and classification; this choice will depend in part on additional developments in the field and work by outside collaborators.

Note that to include LARS options in the full range of mixed effects and classification models is beyond the scope of this project; in particular, some of our algorithms in those areas perform maximum-likelihood estimation using Monte Carlo simulation, and to combine Monte Carlo-based fitting with LARS has not even been considered yet in the research community, to the best of our knowledge.

(Some text is omitted from this version of the proposal.)

Final phase The primary goal for the final phase is to finish things off well—to refine the library and interfaces based on feedback from users, and finish case studies and incorporate them into a manual, finish training materials for short courses. While we create automated tests routinely as part of our normal development, we anticipate additional testing at this stage. Time permitting, we may add additional algorithms, e.g. for a broader range of mixed effects models.

Role of Interns We are budgeting for graduate student and undergraduate interns.

The graduate interns will perform a variety of tasks, depending on their skills, including algorithm development, incorporation of algorithms into the software package, statistical validation, and creating case studies. Our plan is to seek relatively advanced graduate students who have begun research in the regularized regression and classification *(Some text is omitted from this version of the proposal.)* We will place a premium on at least one of the graduate interns having good writing skills, in order to make a strong contribution to the manual, case studies, and training materials.

The undergraduate students will perform a variety of tasks, depending on their skills. Note that our intent is not to have them do grunt work (though there may be some), but to perform more substantive tasks that they find challenging and rewarding. For example, least angle methods are not so complicated to prohibit their use in undergraduate courses; one undergraduate could work with Hesterberg to create a document for use by undergraduate teachers and their students to supplement textbooks. Another task would be to run simulations, to help quantify the performance of these methods in a variety of situations—to help answer the signal-to-noise ratio question. *(Some text is omitted from this version of the proposal.)* We will place a premium on at least one of the undergraduate interns having good writing skills.

The PI has had good experiences with undergraduate interns on two other SBIR-funded projects; in one we created a bootstrap chapter for introductory statistics texts (Hesterberg et al., 2003, 2005), see `www.insightful.com/Hesterberg/bootstrap`; the other for simulation-based econometric software, see `www.insightful.com/Hesterberg/econometrics`. In both cases, the undergraduates had ownership of longer-term tasks, rather than day-to-day assignments.

References

J. L. Adams. A computer experiment to evaluate regression strategies. In *Proceedings of the Statistical Computing Section*, pages 55–62. American Statistical Association, 1990.

N. R. Draper and H. Smith. *Applied regression analysis*. Wiley, third edition, 1998.

Brad Efron and Trevor Hastie. *LARS software for R and Splus*, 2003. `http://www-stat.stanford.edu/~hastie/Papers/LARS`.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–451, 2004.

M.A. Efroymson. Multiple regression analysis. In A. Ralston and H.S. Wilf, editors, *Mathematical Methods for Digital Computers*, volume 1, pages 191–203. Wiley, 1960.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *J. Comput. System Sci.*, 55:119–139, 1997.

Wenjiang Fu. *S-PLUS package brdgrun for shrinkage estimators with bridge penalty*, 2000. URL `http://lib.stat.cmu.edu/S/brdgrun.shar`. `http://lib.stat.cmu.edu/S/brdgrun.shar`.

G. M. Furnival and R. W. Wilson, Jr. Regression by leaps and bounds. *Technometrics*, 16:499–511, 1974.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, October 1999.

Jiang Gui and Hongzhe Li. Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data. *Bioinformatics*, 21:3001–3008, 2005.

Trevor Hastie and Robert Tibshirani. *Generalized Additive Models*. Chapman and Hall, London, 1990.

Trevor Hastie, Rob Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer Verlag, 2001.

Trevor Hastie, Saharon Rosset, Rob Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. 3/5/04, 2004.

Tim Hesterberg, Shaun Monaghan, David S. Moore, Ashley Clipson, and Rachel Epstein. *Bootstrap Methods and Permutation Tests*. W. H. Freeman, 2003. Chapter for *The Practice of Business Statistics* by Moore, McCabe, Duckworth, and Sclove.

Tim Hesterberg, David S. Moore, Shaun Monaghan, Ashley Clipson, and Rachel Epstein. *Bootstrap Methods and Permutation Tests*. W. H. Freeman, second edition edition, 2005. Chapter for *Introduction to the Practice of Statistics* by Moore and McCabe.

C. M. Hurvich and C-L. Tsai. The impact of model selection on inference in linear regression. *The American Statistician*, 44:214–217, 1990.

Hemant Ishwaran. Discussion of “least angle regression” by Efron et al. *Annals of Statistics*, 32(2): 452–458, 2004.

W. J. Kennedy and J. E. Gentle. *Statistical Computing*. Marcel Dekker, New York, 1980.

Jinseog Kim, Yuwon Kim, and Yongdai Kim. Gradient LASSO algorithm. Technical report, Seoul National University, 2005a. URL `http://idea.snu.ac.kr/Research/papers/GLASSO.pdf`.

Jinseog Kim, Yuwon Kim, and Yongdai Kim. *glasso: R-package for Gradient LASSO algorithm*, 2005b. URL `http://idea.snu.ac.kr/Research/glassojskim/glasso.htm`. R package version 0.9, `http://idea.snu.ac.kr/Research/glassojskim/glasso.htm`.

Keith Knight. Discussion of "least angle regression" by Efron et al. *Annals of Statistics*, 32(2):458–460, 2004.

Kenneth Lange. *Numerical Analysis for Statisticians*. Springer-Verlag, New York, 1999.

Justin Lokhorst, Bill Venables, and Berwin Turlach. *Lasso2: L1 Constrained Estimation Routines*, 1999. URL `http://www.maths.uwa.edu.au/~berwin/software/lasso.html`. `http://www.maths.uwa.edu.au/˜berwin/software/lasso.html`.

Jean-Michel Loubes and Pascal Massart. Discussion of "least angle regression" by Efron et al. *Annals of Statistics*, 32(2):460–465, 2004.

David Madigan and Greg Ridgeway. Discussion of "least angle regression" by Efron et al. *Annals of Statistics*, 32(2):465–469, 2004.

Peter McCullagh and John A. Nelder. *Generalised Linear Models*. Chapman and Hall, London, 1989.

Alan Miller. *Subset Selection in Regression*. Chapman & Hall, second edition, 2002.

John F. Monahan. *Numerical Methods of Statistics*. Cambridge University Press, 2001.

M. R. Osborne, Brett Presnell, and Berwin A. Turlach. A new approach to variable selection in least squares problems. *IMA J. Numeri. Anal.*, 20:389–403, 2000.

Mee Young Park and Trevor Hastie. L1 regularization path algorithm for generalized linear models. Unpublished, 2006a. URL `http://www.stanford.edu/~mypark`.

Mee Young Park and Trevor Hastie. *glmpath: L1 Regularization Path for Generalized Linear Models and Proportional Hazards Model*, 2006b. URL `http://cran.r-project.org/src/contrib/Descriptions/glmpath.html`. R package version 0.91.

E. B. Roecker. Prediction error and its estimation for subset-selected models. *Technometrics*, 33:459–468, 1991.

Saharon Rosset. Following curved regularized optimization solution paths. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1153–1160, Cambridge, MA, 2005. MIT Press. URL `http://books.nips.cc/nips17.html`.

Saharon Rosset and Ji Zhu. Discussion of "least angle regression" by Efron et al. *Annals of Statistics*, 32 (2):469–475, 2004a.

Saharon Rosset and Ji Zhu. Piecewise linear regularized solution paths. submitted, 2004b.

Volker Roth. The generalized LASSO. *IEEE Transactions on Neural Networks*, 15:16–28, 2004.

Mark R. Segal, Kam D. Dahlquist, and Bruce R. Conklin. Regression approaches for microarray data analysis. *Journal of Computational Biology*, 10(3):961–980, 2003.

Robert A. Stine. Discussion of “least angle regression” by Efron et al. *Annals of Statistics*, 32(2):475–481, 2004.

Ronald A. Thisted. *Elements of Statistical Computing*. Chapman and Hall, 1988.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society, Series B*, 67(1):91–108, 2005.

Stefan Van Aelst, Jafar A. Khan, and Ruben H. Zamar. Robust linear model selection based on least angle regression. Technical Report, University of British Columbia, 2005. URL `http://hajek.stat.ubc.ca/~ruben/website/cv/RobLARS.pdf`.

Ming Yuan and Yi Lin. Efficient empirical bayes variable selection and estimation in linear models. *Journal of the American Statistical Association*, 100:1215–1225, 2005.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–68, 2006.

Ji Zhu and Trevor Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3):427–444, 2004.

Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. NIPS 2003 (Neural Information Processing Systems). I don’t know if there was a conference proceedings, 2003.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2):301–320, 2005a.

Hui Zou and Trevor Hastie. *elasticnet: Elastic Net Regularization and Variable Selection*, 2005b. R package version 1.0-3.

Hui Zou, Trevor Hastie, and Rob Tibshirani. On the ”degrees of freedom” of the lasso. `http://stat.stanford.edu/˜hastie/pub.htm`, 2004. URL `http://stat.stanford.edu/~hastie/pub.htm`.