S-PLUS and R package for Least Angle Regression

Tim Hesterberg, Chris Fraley
Insightful Corp.

Abstract

Least Angle Regression is a promising technique for variable selection applications, offering a nice alternative to stepwise regression. It provides an explanation for the similar behavior of Lasso (L_1-penalized regression) and forward stagewise regression, and provides a fast implementation of both. We describe a project for creating an open-source S-PLUS/R package `glars` for generalized least angle regression, extending the `lars` package of Efron and Hastie and `glmpath` of Park and Hastie. We invite outside collaboration, and plan for future versions of the package to provide a framework on which others can build.

Keywords: regression, regularization, L_1 penalty.

1 Introduction

> "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 with improvements in data collection technologies. For example, in microarray analysis, the number of predictors (genes) to be analyzed typically far 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 method they call *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.

This idea has caught on rapidly in the academic community—a google scholar search in March 2006 showed 114 citations of (Efron et al. 2004).

In this article we sketch plans for a collaborative effort between outside contributors and Insightful Corporation. The goal of this project is to produce high-quality software for classification and regression based on LARS.

In Section 2 we give an overview of LARS and its relationship to other regression methods. In Sections 3 and 4 we summarize work in the first phase of this project and plans for future work, respectively.

2 Background

In this section we discuss various methods for regression with many variables. 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 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 LASSO, 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.

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, and variables interchangeably (except we use the latter only when it is clear we are discussing predictors 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. We note where there are differences between linear re-

Table 1: Diabetes Study: 442 patients were measured on 10 baseline variables; a prediction model is desired for the response variable Y, a measure of disease progression one year after baseline. Predictors include age, sex, body mass index, average blood pressure, and six different blood serum measurements. 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.

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

gression and the nonlinear cases.

Stepwise and All-Subsets Regression

We begin our description of various regression methods with stepwise and all-subsets regression, which focus on selecting variables 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, 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. For the diabetes data shown in Table 1, single best predictor is BMI; subsequent variables selected are S5, BP, S1, Sex, S2, S4, and S6.

The process is unstable, in that relatively small changes in the data might cause one variable to be selected instead of another, after which subsequent choices may be completely different.

Variations include backward stepwise regression, which starts with a larger model and sequentially removes variables which contribute least to the fit, and Efroymson's procedure (Efroymson 1960), which combines forward and backward steps.

These 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, limited by a maximum number of best subsets (Furnival and Wilson 1974). 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 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 in which there are many more observations than predictors. 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 matrices $X'X$, $X'Y$, and $Y'Y$. Another is to use the QR decomposition. Cross-products and R can both be computed in a single pass through the data, and in both cases there are efficient updating algorithms for adding or deleting variables. However, QR has better numerical properties. See e.g. (Thisted 1988; Monahan 2001; Miller 2002) 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. 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. 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 and instability of variable selection methods has led to other approaches. Ridge regression (Miller 2002; Draper and Smith 1998), includes all predictors, but with typically smaller coefficients than they would have under ordinary least squares. The coefficients minimizing a penalized sum of squares,

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

where θ is a positive scalar; $\theta = 0$ corresponds to ordinary least-squares regression. 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; these shrink as θ increases. Variables most correlated with other variables are affected most, e.g. S1 and S2 have correlation 0.90.

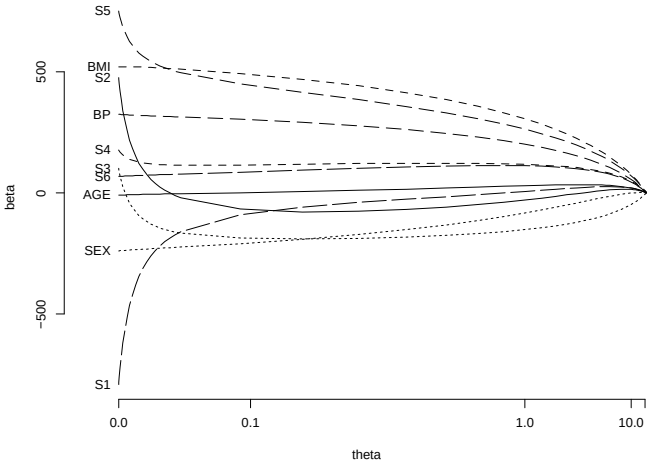

Figure 1: Coefficients for ridge regression (standardized variables)

Note that as θ increases, the coefficients approach but do not equal zero. Hence, no variable is ever excluded from the model (except when 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, which we consider next.

LASSO

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 minimizing the sums of squares of residuals plus an L_1 penalty on the regression coefficients,

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{i,j}\beta_j)^2 + \theta \sum_{j=1}^{p} |\beta_j|. \qquad (3)$$

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.

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. This is 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. Also, the S3 coefficient changes direction when S4 enters the model, ultimately changing sign. This is due to high (negative) correlation between S3 and S4.

Forward Stagewise

Another procedure, forward stagewise regression, 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 the coefficient jumps from zero to the the least-squares value.

Forward stagewise picks the same first variable as forward stepwise, but changes the corresponding coefficient only a small amount. It then picks the variable with highest correlation with the current residuals (possibly 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. The resulting coefficients are more stable than those for stepwise.

Curiously, an idealized version of forward stagewise regression (with the step size tending toward zero) has very similar behavior to the LASSO despite the apparent differences. In the diabetes example, the two methods give identical results until the eighth variable enters, after which there are small differences Efron et al. 2004.

There are also strong connections between forward stagewise regression and the boosting algorithm popular in machine

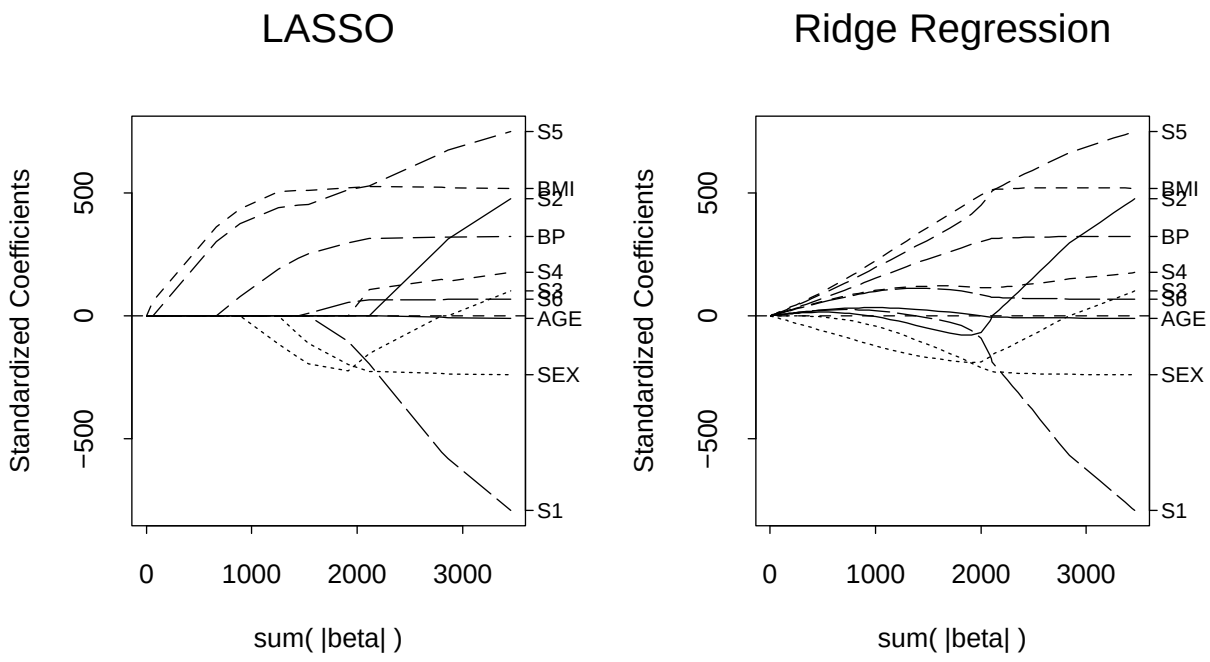

Figure 2: Coefficients for LASSO and Ridge Regression (L_1 and L_2 penalties).

learning (Efron et al. 2004; Hastie et al. 2001). The difference is not in the fitting method, but rather in the predictors used; in stagewise the predictors are typically determined in advance, while in boosting the next variable is typically determined on the fly.

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, the method jumps 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 is greater than

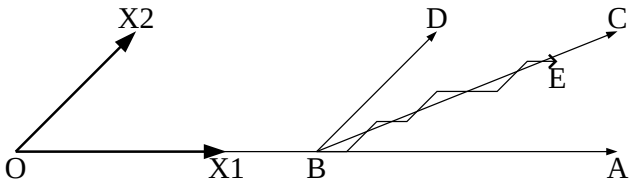

Figure 3: The LAR algorithm in the case of 2 predictors. O is the prediction based solely on an intercept. $C = \hat{Y} = \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$ is the ordinary least-squares fit, the projection of Y onto the subspace spanned by X_1 and X_2. A is the forward stepwise fit after one step; 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 $\to 0$ for stagewise). In higher dimensions additional conditions are needed for exact agreement to hold.

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,

$$C_p = (1/\hat{\sigma}^2) \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 - n + 2k \qquad (4)$$

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.

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.

Related Work

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 (2005b) 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.

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 2006b), generalized linear models (Madigan and Ridgeway 2004; Park and Hastie 2006b), robust linear regression (Rosset and Zhu 2004a; Van Aelst et al. 2005), exponential family models (Rosset 2005), and support vector machines (Zhu et al. 2003; Hastie et al. 2004).

Some additional authors discuss general strategies for solutions in nonlinear models. Roth (2004) discusses a method for iteratively reweighted least squares (IRLS) applications. Ros-

set 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. (2005b) propose a gradient approach particularly useful for high dimensions.

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 that are nearly tied in importance. 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.

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 LARS 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.

3 Phase I Work

Our work on this project falls into two phases of NIH funding—Phase I to demonstrate proof-of-concept was completed in March 2006, and Phase II for more substantial development is just beginning.

There were three technical goals in Phase I:

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

These goals were achieved; we omit details for reasons of space. We also made progress in dealing with linear dependence, and speed improvements.

The most striking aspect of Phase I was our decision to produce an open-source library that will run in both S-PLUS and R, rather than a closed-source version for S-PLUS only. 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.

LARS is an area of active research. 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 2006a), `elasticnet` (Zou and Hastie 2005a), `glasso` (Kim et al. 2005a)).

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 entered 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.

This decision was greeted enthusiastically by key potential collaborators, and a large number of researchers in the area have requested the prototype library produced during Phase I.

Prototype Library: S+GLARS We created a software library 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 2006b). 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).

4 Phase II

This is a rapidly developing field, with the possibility of substantial outside collaboration from academics. Hence our initial efforts will be focused on creating an attractive platform for collaborative work.

The Phase I prototype is not easily extendable. Our goal is to create a framework, in the form of an S-PLUS/R library, 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 provide an 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).

Collaborators may also provide routines for plotting, diagnostics, or other computations.

After the first release of the platform, we plan to refine it based on feedback from collaborators, and to implement extensions such as better support for factors, polynomials and splines, additional types of regression models, other penalty methods such as elastic net (important for microarray data) and fused lasso (important for proteomic data), large-data versions, and missing data handling.

Further development to create a commercial-quality product includes more extensive testing, better documentation, development of case studies, a graphical interface, and interfaces to additional software such as S+ARRAYANALYZER and BIOCONDUCTOR.

Interns The lines between Insightful personnel and outside collaborators may be blurred in one way — we have budgeted for interns. This project should be particularly interesting to graduate students doing research in regularized regression and classification. Anyone interested is invited to contact the authors.

5 Conclusion

We close on 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)."

The procedure has enormous potential, and the goal of this project is to help realize that potential and bring the methodology to the broader statistical community.

For current information please see the project webpage `www.insightful.com/Hesterberg/glars`.

Acknowledgments

This work builds on software by Trevor Hastie, Brad Efron, and Mee Young Park. We thank our consultants Mark Segal, Ji Zhu, and Saharon Rosset, as well as a number of others who provided feedback on Phase I work and suggestions for Phase II.

This work was supported by NIH under NIH SBIR Phase I 1R43GM074313-01 and Phase II 2R44GM074313-02 awards.

References

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

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

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least Angle Regression. *Annals of Statistics*, 32(2):407–451.

Efroymson, M. (1960). Multiple Regression Analysis. In Ralston, A. and Wilf, H., editors, *Mathematical Methods for Digital Computers*, volume 1, pages 191–203. Wiley.

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

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

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

Gui, J. and Li, H. (2005). Penalized Cox Regression Analysis in the High-Dimensional and Low-sample Size Settings, with Applications to Microarray Gene Expression Data. *Bioinformatics*, 21:3001–3008.

Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. (2004). The Entire Regularization Path for the Support Vector Machine. 3/5/04.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer Verlag.

Ishwaran, H. (2004). Discussion of "Least Angle Regression" by Efron et al. *Annals of Statistics*, 32(2):452–458.

Kim, J., Kim, Y., and Kim, Y. (2005a). *glasso: R-package for Gradient LASSO algorithm*. R package version 0.9, `http://idea.snu.ac.kr/Research/glassojskim/glasso.htm`.

Kim, J., Kim, Y., and Kim, Y. (2005b). Gradient LASSO algorithm. Technical report, Seoul National University.

Knight, K. (2004). Discussion of "Least Angle Regression" by Efron et al. *Annals of Statistics*, 32(2):458–460.

Lokhorst, J., Venables, B., and Turlach, B. (1999). *Lasso2: L1 Constrained Estimation Routines*. `http://www.maths.uwa.edu.au/~berwin/software/lasso.html`.

Loubes, J.-M. and Massart, P. (2004). Discussion of "Least Angle Regression" by Efron et al. *Annals of Statistics*, 32(2):460–465.

Madigan, D. and Ridgeway, G. (2004). Discussion of "Least Angle Regression" by Efron et al. *Annals of Statistics*, 32(2):465–469.

Miller, A. (2002). *Subset Selection in Regression*. Chapman & Hall, second edition.

Monahan, J. F. (2001). *Numerical Methods of Statistics*. Cambridge University Press.

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

Park, M. Y. and Hastie, T. (2006a). *glmpath: L1 Regularization Path for Generalized Linear Models and Proportional Hazards Model*. R package version 0.91.

Park, M. Y. and Hastie, T. (2006b). L1 Regularization Path Algorithm for Generalized Linear Models. Unpublished.

Rosset, S. (2005). Following Curved Regularized Optimization Solution Paths. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 1153–1160, Cambridge, MA. MIT Press.

Rosset, S. and Zhu, J. (2004a). Discussion of "Least Angle Regression" by Efron et al. *Annals of Statistics*, 32(2):469–475.

Rosset, S. and Zhu, J. (2004b). Piecewise Linear Regularized Solution Paths. submitted.

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

Segal, M. R., Dahlquist, K. D., and Conklin, B. R. (2003). Regression Approaches for Microarray Data Analysis. *Journal of Computational Biology*, 10(3):961–980.

Stine, R. A. (2004). Discussion of "Least Angle Regression" by Efron et al. *Annals of Statistics*, 32(2):475–481.

Thisted, R. A. (1988). *Elements of Statistical Computing*. Chapman and Hall.

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

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society, Series B*, 67(1):91–108.

Van Aelst, S., Khan, J. A., and Zamar, R. H. (2005). Robust Linear Model Selection Based on Least Angle Regression. Technical Report, University of British Columbia.

Yuan, M. and Lin, Y. (2006). Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–68.

Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. (2003). 1-norm Support Vector Machines. NIPS 2003 (Neural Information Processing Systems). I don't know if there was a conference proceedings.

Zou, H. and Hastie, T. (2005a). *elasticnet: Elastic Net Regularization and Variable Selection*. R package version 1.0-3.

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

Zou, H., Hastie, T., and Tibshirani, R. (2004). On the "Degrees of Freedom" of the Lasso. `http://stat.stanford.edu/~hastie/pub.htm`.