Simon Fraser University
Faculty of Statistics & Actuarial Science
Burnaby, British Columbia

Least Angle Regression

Presented By:

Barinder Thind

Gabriel Phelan

Contents

1 Introduction

In many modern statistical analyses, data sets include a plethora of covariates from which some subset results in an "optimal" model; indeed, if the number of covariates exceeds the number of observations the data analyst is forced to confront this issue. The question of which predictors to include for regression is often difficult to answer; as a remedy to this problem, a number of selection algorithms have been proposed that come packaged with their own benefits and drawbacks. In particular, the stepwise and stagewise feature selection techniques serve as tools to pick which variables to include and, in the case of the latter, *how much* of each to include. However, these methods have disadvantages that can often stifle their utility - enter, Least Angle Regression (LARS); this algorithm uses geometry to simultaneously attain the computational efficiency of stepwise selection and the statistical efficiency of the stagewise algorithm. It also provides an enlightening connection to the popular LASSO algorithm, effectively unifying the zoo of forward selection procedures. This paper details each of these approaches and presents the mathematics fundamental to the relative superiority of LARS.

In Section II, we introduce the methodology underpinning the aforementioned selection algorithms. In Section III, we highlight the predictive power of each algorithm along with runtime comparisons. In Section IV & Section V, future considerations and references are provided. Lastly, the Appendix has solutions to some interesting problems associated with the selection methods outlined in this report.

2 Methodology

In this section, we introduce the mathematics underpinning LARS and highlight its connection with other approaches to subset selection, such as forward stepwise regression, forward stagewise regression, and the LASSO. In the case of LARS, the mathematics is not merely a precursor to understanding how the algorithm works. Rather, it is at the core of its appeal. LARS' greatest contribution is the unifying and elegant framework in which it casts subset selection, offering new insights into the range of methods currently in use. In this way, the current section is imperative to our presentation. We conjecture that the reader who deeply understands the theory of LARS but remembers nothing of its implementation will benefit more than the reader for whom the opposite is true. We assume the usual setup throughout, that is, we have a response $y \in \mathbb{R}^n$ and a matrix of covariates $X \in \mathbb{R}^{n \times p}$ related via the linear model $y = X\beta, \beta \in \mathbb{R}^p$. Our goal is variable selection – we think some columns of X exert more influence on y than others and wish to find $\hat{y} = \hat{X}\hat{\beta}$ with only these "best" predictors included. We take both y and the columns of X to be centered; the columns of X are also scaled to unit-length. Any correlations we encounter, which will arise in the form of inner products, are assumed to be positive. Doing away with this assumption (as we will of course do during implementation) is straightforward but clutters the mathematics and in our opinion adds no further understanding. It is useful to have in mind the geometric interpretation of multiple regression, as there will be much interplay between geometry and algebra in what follows.

2.1 Stepwise Regression

The first step in understanding LARS is via the simpler forward stepwise procedure [13]. This section serves to establish the setting from which other algorithms will emerge. As with LARS, the idea is to build up the model successively, adding in predictors until we are satisfied. We formalize this as follows. First, note that the correlation [4] between y and some feature vector x_i is given by

$$\frac{\sum_{j=1}^{n} x_{ij} y_j}{\sum_{j=1}^{n} x_{ij}^2 \sum_{j=1}^{n} y_j^2} = \frac{\langle x_i, y \rangle}{||y||_2^2} \tag{1}$$

where this expression is greatly simplified by the centering and scaling that we assumed above. Thus, if we wanted to find the predictor most correlated with y, we could compute $X^T y$, who's i^{th} entry is proportional to the result in 1. Suppose that upon doing this, we indeed found x_i to be maximally correlated with y; x_i would then be the first predictor added to our model, with coefficient equal to the usual OLS coefficient. If we call this one-predictor model as $\hat{y}$, we may define the residual vector as $r = y - \hat{y}$, i.e., the difference between the true response and the response as predicted by our submodel. The second term to enter the model is determined by the covariate most correlated with r, again with the OLS coefficient. This repeats, at each step updating $\hat{y}$ and r. Put another way, at each iterate stepwise finds the predictor that most improves the model fit. A succinct way of summarizing the algorithm is

$$\hat{y} \leftarrow \hat{y} + \hat{\beta}_{\text{OLS}} x \tag{2}$$

for $\hat{\beta}_{\text{OLS}} x$ the most recently added term. Of course, because we add features with their OLS coefficients, without a halting criterion we eventually arrive at the usual least-squares solution. Thus, an extremely naive way to implement stepwise regression would involve computing and storing the least-squares estimates $\left\{\hat{\beta}^1_{OLS}, \ldots, \hat{\beta}^p_{\text{OLS}}\right\}$, then iterating as above. The shortcomings of this approach are obvious. Presumably if we are doing variable selection, we *don't want* the full OLS solution, thus finding it is computationally wasteful. This may matter greatly in high-dimensional problems. What's more, if $p > n$ this approach isn't even possible. What we want then is a way to compute the coefficients as we run through the algorithm; orthogonalization will provide the solution [13, 11].

Recall Gram-Schmidt orthogonalization [12], which states that any linearly independent set of vectors spawns an orthogonal basis for the space spanned by the original set. This is achieved via the Gram-Schmidt algorithm. We omit the specifics (see [13] for a fuller treatment), but the utility of Gram-Schmidt in our context is that like stepwise, it is an iterative procedure. The orthogonalization occurs one by one, via projections onto the span of those vectors *already* orthogonalized. This plugs seamlessly into stepwise. We simply apply Gram-Schmidt as we go, so that after each iterate we know both

$$\mathcal{I} = \{\text{those } x_i \text{ already in the model}\} \tag{3}$$

and

$$\mathcal{O} = \{z_i, \text{ the orthogonalized } x_i \in \mathcal{I}\}. \tag{4}$$

We claim that this allows for computation of the OLS solutions *as we go*. Consider Z, the matrix who's columns are the orthogonalized inputs. Regressing y on Z, the least-squares estimate is

$$\hat{\beta} = (Z^T Z)^{-1} Z^T y. \tag{5}$$

By our assumptions, the columns of Z are not just orthogonal but *orthonormal* as well, thus this reduces to $\hat{\beta} = Z^T y$. That is, the i^{th} OLS coefficient is given by $\beta_i = \langle z_i, y \rangle$ – the same as if we'd simply ran a univariate procedure regressing y on z_i [13]. The remarkable fact is that this is *the same* as the OLS coefficient for the unorthogonalized input x_i. To see this, recall that in ordinary linear regression $\hat{y}$ is the projection of y onto the column space of X. By Gram-Schmidt, the column space of X is the same as the column space of Z, so it must be that $\hat{\beta}$ is the same in both cases. This gives an extremely simple way to update in the midst of a stepwise procedure; we now have

$$\hat{y} \leftarrow \hat{y} + \langle z, y \rangle x. \tag{6}$$

To halt the above scheme, thus choosing a final subset of predictors, one has options. Clearly, if $p > n$ one must stop when p covariates have entered the model. Alternatively, we could choose a priori the number of covariates to include, or, let cross validation choose for us.

2.2 Stagewise Regression

The reader may notice that the forward stepwise algorithm is extremely greedy – we make optimal[1] decisions at each step of the algorithm but without regard for the overall optimality. Forward stagewise regression [8, 13] is an attempt to remedy this by adding variables to the model in increments, rather than going "all-in" as stepwise

[1] We use the term *optimal* in a very loose sense throughout the paper.

does. We need only modify 6 slightly to obtain the stagewise update. Suppose we prespecify some small $\varepsilon > 0$. We proceed as before, but add only a fraction ε of the most-correlated predictor at a time. In short, the update becomes

$$\hat{y} \leftarrow \hat{y} + \varepsilon x \tag{7}$$

This means that x is added bit-by-bit *until* some other variable becomes more correlated with the residual. Put another way, we will be able to detect (up to discretization error) precisely when a predictor becomes more correlated with r than the one we are currently incrementing, and begin to add that predictor instead. Note that taking $\varepsilon = \langle z, y \rangle$ simply gives stepwise update. Roughly speaking, we only add as much of a predictor as is needed, thus stagewise has been called a more democratic version of stepwise [13, 9]. Of course, the major drawback of stagewise is its lack of efficiency. Because we now update in ε-increments, the number of iterations explodes. Stagewise sometimes requires in the thousands of iterations [8] to arrive at a satisfactory solution; a major motivation of LARS is to improve upon this situation.

2.3 Least Angle Regression

We now have the necessary tools to understand LARS [8]. From a high-level point of view, LARS tries to marry the efficiency of stepwise with intelligent update rule of stagewise. As it turns out, this is achieved via a beautiful appeal to geometry which provides new perspectives on the family of forward selection algorithms and the seemingly-unrelated LASSO.

Fortunately, the ambitious goals of LARS can indeed be achieved. To understand how, consider a toy example with only two predictors x_1, x_2. Suppose that we initially find $\langle x_1, y\rangle > \langle x_2, y\rangle$. Then classical stagewise iterates $\hat{y} \leftarrow \hat{y} + \varepsilon x_1$ until x_2 becomes equally correlated with the current residual, which occurs at say $\hat{y} = \gamma_1 x_1$. This is where "LARS parts company with forward selection" [8]. Rather than add a multiple of x_2 to the model, LARS now adds a multiple of a vector v that is is equiangular (this is where the name derives from) to x_1 and x_2 until our estimate for y looks like

$$\hat{y} = \gamma_1 x_1 + \gamma_2 v. \tag{8}$$

Of course, v depends on x_1, x_2, so in the end we still end up with a linear combination of our feature vectors. The extension of this idea to higher dimensions is immediate. Aside from being an elegant idea in its own right, the amazing thing about this construction is that we can actually compute the optimal step sizes $\gamma_1, \gamma_2, \ldots$ at each step of the algorithm, thereby eliminating the tiny increments of stagewise. We thus improve upon stagewise's democratic variable-selection, but like stepwise, a model with q terms requires only q iterates of the algorithm.

With the intuition of LARS established, we now show how to compute the vector v equiangular to $\{x_1, \ldots, x_p\} \in \mathbb{R}^n$ [1], as well as the step sizes $\gamma_1, \gamma_2 \ldots$ that give LARS its efficiency. Recall that for vectors v, x_i,

$$\langle v, x_i\rangle = ||v||_2 \, ||x_i||_2 \cos\varphi \tag{9}$$

where φ is the angle between v, x_i in $\mathbb{R}^n$. The fact that $||x_i||_2 = 1$ for all i implies that v is equiangular to x_i and x_j if and only if $\langle v, x_i\rangle = \langle v, x_j\rangle$. We can succinctly extend this idea to the entire set $\{x_1, \ldots, x_p\}$ by requiring

$$X^T v = \begin{bmatrix} \alpha \\ \vdots \\ \alpha \end{bmatrix} \tag{10}$$

for some constant α. It will be convenient to take $\alpha = 1$, so our goal is to find v satisfying $X^T v = j$ for j the vector of ones. For simplicity, we will also seek $||v||_2 = 1$. Rather than attempting to brute-force our way to a solution, we can take a more constructive approach. As a trivial observation, we have $Ij = j$. This suggests finding some matrix A depending on X^T so as to write $AA^{-1}j = j$. One quickly realizes that $A = X^T X$ fits our criterion, i.e.,

$$X^T \underbrace{X(X^T X)^{-1} j}_{v} = j \tag{11}$$

which gives us v up to a constant. Scaling to unit-length:

$$\begin{aligned}
\left\|X(X^T X)^{-1} j\right\|_2 &= \left\langle X(X^T X)^{-1} j, X(X^T X)^{-1} j \right\rangle^{1/2} && (12)\\
&= \left[\left(X(X^T X)^{-1} j\right)^T X(X^T X)^{-1} j\right]^{1/2} && (13)\\
&= \left[j^T \left(X^T X\right)^{-1} X^T X(X^T X)^{-1} j\right]^{1/2} && (14)\\
&= \left[j^T (X^T X)^{-1} j\right]^{1/2} && (15)
\end{aligned}$$

and so we get

$$v = X(X^T X)^{-1} j \Big/ \left[j^T (X^T X)^{-1} j\right]^{1/2}. \tag{16}$$

To summarize, the above vector is of unit-length and is equiangular to each the columns of X. Clearly, the fact we can find an explicit formula for this vector is a boon to the successful implementation of the algorithm.

One may begin to wonder just why this equiangular vector is so special. As eluded to before, its geometry leads to straightforward calculations of the step sizes $\gamma_1, \gamma_2, \ldots$; we demonstrate this now. Suppose that we have just completed a step of LARS and that $\hat{y} + \gamma v$ is our current model. In words, γv has just been added to the previous model $\hat{y}$. Define $\mathcal{I}$ as in 3 and compute the correlation between r and x_i for some $x_i \in \mathcal{I}$:

$$\langle x_i, r\rangle = \langle x_i, y - (\hat{y} + \gamma v)\rangle \tag{17}$$

$$= \langle x_i, y - \hat{y}\rangle - \gamma\langle x_i, v\rangle. \tag{18}$$

If we may be so bold, equation 18 holds the key to the entire algorithm. It states that the current correlation between x_i and r is equal to the correlation from the previous step minus a multiple of $\langle x_i, v\rangle$. But, since v makes equal angles with all vectors currently in the model, $\langle x_i, v\rangle$ *is the same for all* $x_i \in \mathcal{I}$ and we call this value d. So, the correlation decreases at an equal rate γd among all the x_i's currently in the model. Furthermore, because $\langle x_i, y - \hat{y}\rangle$ is just the residual from the algorithm's last step, an inductive argument shows that it *also* is the same for all $x_i \in \mathcal{I}$. The ultimate conclusion is that the correlations of all "active" [8] predictors $\{\langle x_i, r\rangle\}_{i\in\mathcal{I}}$ is the same; call this value c.

We now use this fact in deriving the step size γ_k incurred on step k of the algorithm. Similarly, we append the subscript k to all quantities previously discussed to indicate their current state. For active predictors, the discussion above can be compactly written as

$$c_k = c_{k-1} - \gamma_k d_k. \tag{19}$$

We want γ_k so that the predictor $x_j \in \mathcal{I}^c$ that we add to the model at step $k+1$ has the property $\langle x_j, r_k \rangle = c_k$ (this is just our criterion for adding in predictors). Expanding this out,

$$\langle x_j, r_k \rangle = c_k \tag{20}$$

$$\langle x_j, r_{k-1} \rangle - \gamma_k \langle x_j, v_k \rangle = c_{k-1} - \gamma_k d_k \tag{21}$$

$$\gamma_k = \frac{c_{k-1} - \langle x_j, r_{k-1} \rangle}{d_k - \langle x_j, v_k \rangle} \tag{22}$$

where we have used the results derived above and elementary algebra. Of course, we don't know which x_j where are going to add next, but making γ_k as small as possible so that *some* such x_j enters the model means the general rule is

$$\gamma_k = \min_{x_j \in \mathcal{I}^c} \left\{ \frac{c_{k-1} - \langle x_j, r_{k-1} \rangle}{d_k - \langle x_j, v_k \rangle} \right\}. \tag{23}$$

Remarkably, all the above quantities are known and computable at time k. We also point out that LARS indeed recovers the OLS solution if left to exhaust the set of predictors (assuming this is possible) [8]; We defer readers to Efron et. al. for the mathematical details of this fact.

As we hope is evident from the preceding treatment, LARS is essentially a much cleverer version of stepwise and stagewise regression. That it makes its ancestors seem like simplistic approximations speaks to its sophistication. Exploiting the geometric and linear algebraic nature of the problem, LARS expounds what these older approaches “should” have been doing all along. In this sense it is the ultimate forward selection algorithm, at once effective, efficient, and elegant.

2.4 LASSO from LARS

Perhaps LARS’ greatest trick is its surprising connection to the LASSO [8, 10]. Despite being motivated from completely different perspectives (forward selection in the case of LARS, convex optimization in the case of LASSO), a simple modification of LARS reveals deep connections between these two methods. Recall that LASSO solves

$$\underset{\beta}{\arg\min} \, ||y - X\beta||_2 + \lambda \, ||\beta||_1 \, , \tag{24}$$

the 1-norm inducing sparsity. Often one wants to solve this optimization problem for all values of the tuning parameter λ. Pre-LARS, this would proceed by repeatedly solving a quadratic program [8]. Remarkably, modifying LARS only slightly allows for this to be computed in one pass. The modification is this: if the correlation between a feature and the residual crosses 0, drop it from computation of the equiangular vector v [9]. In short, this gives a more efficient LASSO solution. Interestingly, another modification to LARS gives the stagewise path as $\varepsilon \to 0$ [7].

3 Results

3.1 Data Description

There were three datasets used for this paper - one was generated for the purpose of runtime comparison, another for variable path comparison, and a final dataset was used to contrast predictive accuracy. For the first, 3 data subsets were randomly generated with values taken from a normal distribution with some fixed mean and variance[2]. Computations were performed via the lars package in R [3].

Dataset Number	Dimension (n x p)
1	500 x 100
2	600 x 1000
3	700 x 10000

Table 1: Runtime Datasets

The data used to asses variable selection is due to LLoyd Elliott and was created for use in Kaggle competitions [2]. With 75 covariates and many internal pathologies, this dataset was designed to be a tricky exercise in big data prediction. For the mean squared error comparisons, a prostate data set was used [6]. There were 9 variables and 97 observations, 60 of which were used in a training set, the rest testing.

3.2 Comparison

3.2.1 Runtimes

The runtimes associated with each of the selection methods are outlined below in table 2.

Dataset Number	Stepwise	Stagewise	LARS	LASSO
1	0.0422 s	0.0493 s	0.0417 s	0.0334 s
2	3.9400 s	22.245 s	2.9655 s	7.8207 s
3	1.1434 m	1.8870 m	1.3181 m	2.2626 m

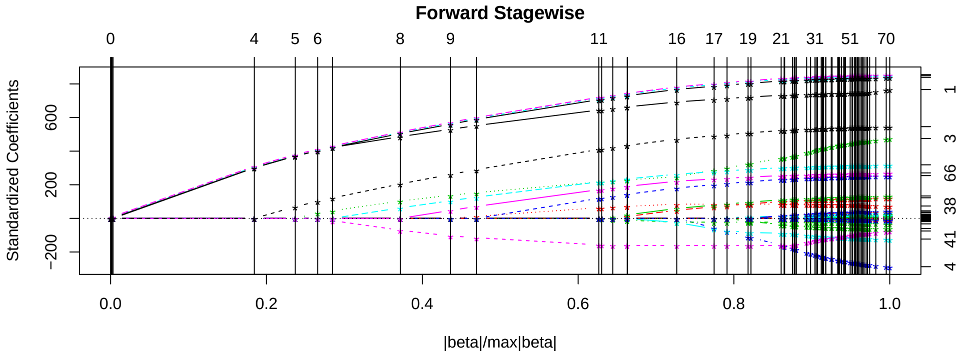
Table 2: Runtime Comparisons

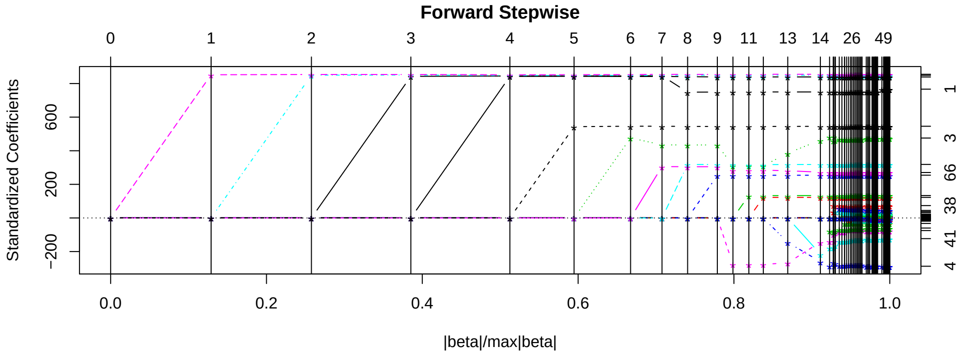
As expected, the differences in runtime for the smaller dataset are negligible. The difference becomes more pronounced in the second dataset where stagewise is significantly slower than LARS, LASSO, and stepwise. This pattern only continues as the size of the dataset increases. With this, we demonstrate that LARS indeed

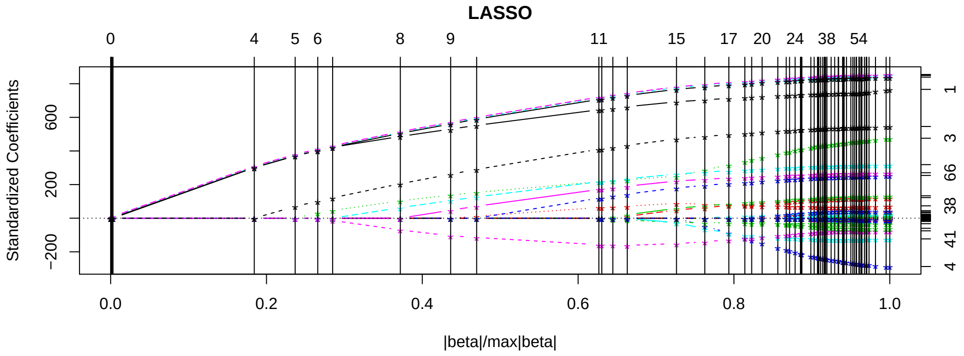
[2] The particular values here are not important as the purpose was to test runtimes for this data

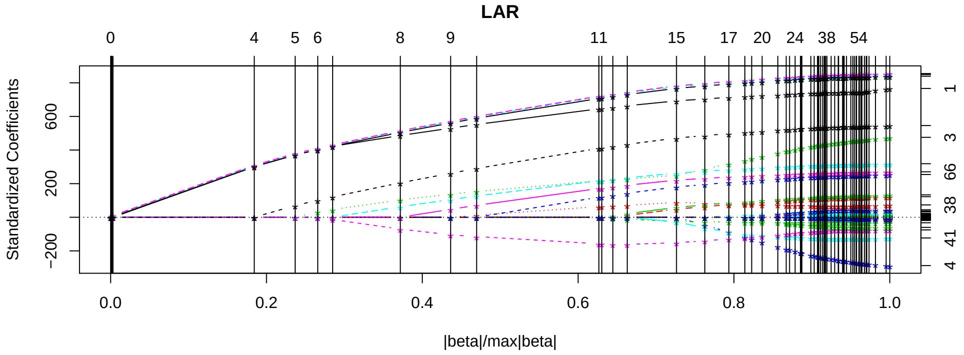
maintains efficiency when compared to its forward selection cousins.

3.2.2 Paths









See above the variable paths for the selection techniques here discussed. The algorithms iterate left to right, adding predictors as needed (each line in the plot corresponds to a predictor, the height corresponding to its coefficient). Stepwise selection immediately stands out, with its large, predictable jumps. Note how similar the results of the other three algorithms are; this elucidates their connections. In light of our methodological discussion, this shouldn't be surprising.

3.2.3 Prediction

Despite all its elegance, the success of LARS depends on its actual performance. As we show, it predicts as well, if not better, than the competing algorithms. Even if outperformed (as it inevitably will be on certain data sets), the gains in computational efficiency should still make it an appealing choice. The prostate data set was used to make predictions on the log of the cancer volume in 200 patients. Table 3 provides results over five seeds.

Method/Dataset	Stepwise	Stagewise	LARS	LASSO	Winner
Prediction Set 1:	0.3984	0.4308	0.4318	0.5136	Stepwise
Prediction Set 2:	1.045	1.077	1.042	1.141	LARS
Prediction Set 3:	0.6462	0.6274	0.6275	0.6999	LARS
Prediction Set 4:	0.4987	0.5099	0.5100	0.7184	Stepwise
Prediction Set 5:	0.5923	0.5701	0.5661	0.5833	LARS

Table 3: MSE Comparisons

In order to get a more concrete comparison, a cross-validation approach was taken and, over a 1000 iterations, the averaged MSE's are given in Table 4.

	Stepwise	Stagewise	LARS	LASSO
Averaged MSE:	0.5992	0.5705	0.5710	0.6903

Table 4: Overall MSE Comparisons

The negligible advantage of stagewise notwithstanding, LARS predicts the best of all the algorithms. Recall too the inefficiency of stagewise; LARS would be the only feasible choice in high dimensions if predictive accuracy were the goal.

4 Conclusions & Future Considerations

Model selection is an important part of nearly any analysis. The variables we choose to retain can significantly alter inferences made from the dataset. Prediction too can benefit from variable selection, especially when $p > n$ and the model is singular. Hence, it is important that we pick these variables in as reasonable a manner as possible. Forward selection algorithms like stepwise and stagewise regression have been proposed as solutions to this problem, but they suffer from statistical and computational inefficiency respectively. Least Angle Regression uses geometry to bypass these problems and marries the best of these popular tools. It also provides an unexpected perspective on the LASSO, and gives a more efficient way of computing entire LASSO paths. In this paper, we discussed mathematical justifications for these claims, and demonstrated them on a mix of real and synthetic data. Ultimately, we recommend the usage of LARS for all forward selection-based regression modeling.

With respect to future developments, it seems that the benefits of LARS don't need to be necessarily limited to the multivariate case; an approach involving the discretization of functional observations (in the context of *Functional Data Analysis* [5]) to exploit the benefits of LARS seems to show promise as a research topic.

5 References

[1] https://stats.stackexchange.com/questions/308942/how-to-get-the-equiangular-vector-in-p-dimension-linear-space-used-in-least-ang.

[2] Lloyd Elliott. *Sassafras*. dataset.

[3] Trevor Hastie and Brad Efron. *lars: Least Angle Regression, Lasso and Forward Stagewise*. R package version 1.2. 2013. URL: https://CRAN.R-project.org/package=lars.

[4] Nathaniel E. Hellwig. *Data, Covariance, and Correlation Matrix*. http://users.stat.umn.edu/~helwig/notes/datamat-Notes.pdf.

[5] B.W. Silverman James Ramsay. *Functional Data Analysis*. Springer, 1997.

[6] Justin Lokhorst et al. *lasso2: L1 Constrained Estimation aka 'lasso'*. R package version 1.2-20. 2018. URL: https://CRAN.R-project.org/package=lasso2.

[7] Brad Efron. Trevor Hastie. Iain Johnstone. Robert Tibshirani. *Least Angle Regression, Forward Stagewise, and the LASSO*. https://web.stanford.edu/~hastie/TALKS/larstalk.pdf.

[8] Bradley Efron. Trevor Hastie. Iain Johnstone. Robert Tibshirani. «Least Angle Regression». In: *The Annals of Statistics* 32 (2004), pp. 407–499.

[9] Robert Tibshirani. *A Simple Explanation of the LASSO and Least Angle Regression*. https://statweb.stanford.edu/~tibs/lasso/simple.html.

[10] Robert Tibshirani. «Regression Shrinkage and Selection via the LASSO». In: *J. Royal Stat. Soc.* 58 (1996), pp. 267–288.

[11] Ryan Tibshirani. *Regression 2: More Perspectives, Shortcomings*. http://www.stat.cmu.edu/~ryantibs/datamining/lectures/14-reg2.pdf|.

[12] Sergei Treil. *Linear Algebra Done Wrong*. 2017.

[13] Jerome Friedman Trevor Hastie Robert Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

6 Appendix

6.1 Questions

1. Prove that stepwise regression recovers the OLS solution. That is, show that $\hat{\beta}_i = \langle z_i, y \rangle$ (under the assumptions we gave).

2. An assumption of LARS that we didn't mention was that we need for the x_i's in the model at each step to be linearly independent (otherwise the algorithm will halt). Give a reason specific to the geometry of LARS as to why this must be so (no calculations required – just think about why this has to be the case).

3. Give an explicit formula for the angle φ between the equiangular vector v and a predictor x_i already in the model.

6.2 Solutions

1. This was shown in the paper.

2. If the x_i's are not linearly independent, there doesn't exist an equiangular vector between them. The reader is invited to draw three vectors in the plane, and attempt to find a vector that makes equal angles with all three.

3. This is a straightforward algebraic manipulation.