Forward-LASSO with Adaptive Shrinkage

PETER RADCHENKO AND GARETH M. JAMES *

Abstract

Both classical Forward Selection and the more modern Lasso provide computationally feasible methods for performing variable selection in high dimensional regression problems involving many predictors. We note that although the Lasso is the solution to an optimization problem while Forward Selection is purely algorithmic, the two methods turn out to operate in surprisingly similar fashions. Our results demonstrate, both empirically and theoretically, that neither procedure dominates the other. We propose a new method we call Forward-Lasso Adaptive SHrinkage (FLASH), which incorporates both Forward Selection and the Lasso as special cases. FLASH works well in situations where either Forward Selection or the Lasso dominates but also performs well in situations where neither method succeeds. FLASH is fitted using a variant of the computationally efficient LARS algorithm. We provide an extensive theoretical analysis showing that many of the error bounds that have recently been developed for the Lasso can be improved using FLASH. Finally we demonstrate, through numerous simulations and a real world data set, that FLASH generally outperforms many competing approaches.

Some key words: Forward Selection; Lasso; Shrinkage; Variable Selection

1 Introduction

Consider the traditional linear regression model,

$$Y_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + \epsilon_i, \quad i = 1, \ldots n, \quad (1)$$

with $p$ predictors and $n$ observations. Recently attention has focussed on the scenario where $p$ is large relative to $n$. In this situation there are many methods that outperform ordinary least squares (OLS) (Frank and Friedman, 1993). One common approach is to assume that the true number of regression coefficients, i.e. the number

*Marshall School of Business, University of Southern California. This work was partially supported by NSF Grant DMS-0705312.
of nonzero $\beta_j$’s, is small, in which case estimation results can be improved by first performing variable selection. Many variable selection methods have been proposed. They can be divided into “classical” and “modern” approaches. Among the classical methods one of the simplest and most well known is Forward Selection (FS). There are two common variants of FS. One version starts with a model containing no variables, then at each iteration adds the variable that produces the largest reduction in the sum of squares, conditional on the previously selected variables. The alternative implementation, and the version we concentrate on, also starts with a model containing no variables, then at each iteration adds the variable that has the highest correlation with the current residual vector, $Y - X\hat{\beta}$, where the residuals are computed using an OLS fit on the already selected set of variables. Selecting the variable with highest correlation to the residual vector results in the greatest instantaneous reduction in the sum of squares for a small change in the regression coefficient so the two versions of FS can be seen as approximations to each other. While it is easy to construct theoretical examples where this greedy approach will fail, in practice it often gives good results.

More recently interest has focused on an alternative “modern” class of methods that implement both variable selection and coefficient shrinkage in a single procedure. The most well known of these procedures is the Lasso (Tibshirani, 1996). In addition to minimizing the usual sum of squares, the Lasso imposes an $L_1$ penalty on the coefficients which has the effect of automatically performing variable selection by setting certain coefficients to zero and shrinking the remainder. While, in general, the latter property is very appealing, it has been shown that in sparse settings the Lasso often over shrinks the coefficients. Numerous alternatives and extensions have been suggested. A few examples include SCAD (Fan and Li, 2001), the Elastic Net (Zou and Hastie, 2005), Adaptive Lasso (Zou, 2006), the Dantzig selector (Candes and Tao, 2007), the Relaxed Lasso (Meinshausen, 2007), VISA (Radchenko and James, 2008) and the Double Dantzig (James and Radchenko, 2009).

The Lasso has been made particularly appealing by the advent of the LARS algorithm (Efron et al., 2004) which provides a highly efficient means to simultaneously produce the set of Lasso fits for all values of the tuning parameter. Like FS, the LARS algorithm starts with an empty set of variables and then adds the variable most highly correlated with the response. At this stage the coefficient of the selected variable, say $x_j$, is adjusted until the absolute correlation between $x_j$ and the residual vector is reached by the corresponding absolute correlation of another predictor, say $x_k$. The new predictor is then added to the model and the coefficients for both $x_j$ and $x_k$ are adjusted in such a way as to drive the corresponding absolute correlations down towards zero at the same rate. This procedure continues, adding a new variable each time it’s absolute correlation reaches the maximum, until all correlations have been driven to zero, which corresponds to the least squares solution.

In comparing the FS and Lasso procedures we note that both methods select the next variable to include based on the correlations between predictors and the resid-
uals. The key difference is that FS computes the residuals using the unshrunk least squares coefficients while the Lasso uses coefficient estimates that are shrunk towards zero. Our simulation studies in Section 4 show that on relatively easier data sets, where the correct model is identified more often, the FS approach dominates. However, on harder problems, where more mistakes are made in the choice of variables, the Lasso approach is superior. This paper is based on an observation that the Lasso and Forward Selection can be viewed as two extremes of a continuum of possible model selection rules. Instead of selecting candidate models using either the highly shrunk Lasso approach or the unshrunk FS procedure we propose a methodology that can adaptively adjust the level of shrinkage, not just on the final model coefficients, as used previously in e.g. the Relaxed Lasso, but also during the selection of potential candidate models. We call our approach “Forward-Lasso Adaptive SHrinkage” (FLASH). At each step in the algorithm FLASH automatically decides on the optimal level of shrinkage before choosing the next candidate variable. The level of shrinkage selected will be between that of FS and the Lasso, so that both of these methods are included as special cases. FLASH has the same order of computational cost as the Lasso, but we show through an extensive simulation study as well as theoretical arguments that it significantly outperforms FS, the Lasso, and many alternative methods.

Our paper is structured as follows. In Section 2 we discuss the relationship between FS and the Lasso more carefully. We then introduce FLASH as a compromise between the two methods and present an algorithm for constructing its path. In Section 3 we investigate the variable selection and coefficient estimation properties of FLASH and provide a number of corresponding theoretical results. Then in Section 4 we present a detailed simulation study to examine the practical performance of FLASH in comparison to FS, the Lasso and other competing methods. A real world data set with large $p$ is examined in Section 5 and we end with a discussion in Section 6.

## 2 Methodology

Using suitable location and scale transformations we can standardize the predictors and center the response, so that

\[
\sum_{i=1}^{n} Y_i = 0, \quad \sum_{i=1}^{n} X_{ij} = 0, \quad \sum_{i=1}^{n} X_{ij}^2 = 1, \quad \text{for } j = 1, \ldots, p. \tag{2}
\]

Throughout the paper we assume that (2) holds. However, all numerical results are presented on the original scale of the data.

### 2.1 Lasso Versus Forward Selection

In order to better understand the similarities and differences between the Lasso and FS methods Figure 1 provides a pictorial representation on a hypothetical example...
with just four predictor variables. The top figure plots absolute correlations between predictors and residuals for all possible Lasso solutions. On the left hand side of the plot all coefficients start at zero and the black solid line variable is the most highly correlated with the response, hence the Lasso selects it first. The coefficient for this variable is then allowed to change, and the corresponding absolute correlation is driven towards zero. The process stops, or “breaks”, when the blue dash-dot variable’s absolute correlation catches up to the maximum. The blue variable is then added to the model and both absolute correlations are driven towards zero at the same rate. Next, the green dashed variable is added and finally the red long dash variable before all the absolute correlations reach zero, which corresponds to the least squares solution. We use the term “break” to refer to a point where a new variable enters or leaves the model, and hence the direction of the path changes.

Typically, FS is not thought of in terms of a path. However, one can easily create a path by linearly interpolating the ordinary least squares coefficients for each of the selected models. The lower plot in Figure 1 provides a pictorial representation of the FS absolute correlation path for the same four variable example. As in the top
plot, the black variable is selected first, but the path does not deviate when the blue
and black absolute correlations become equal, instead it continues until the black one
reaches zero. At this point the red variable has the maximum absolute correlation,
then it is selected by FS. Its absolute correlation is driven to zero, while maintaining
the black correlation at zero. The blue variable is selected next, followed by the green
variable, and their absolute correlations are driven to zero as well.

In this hypothetical example the Lasso and FS sequences of selected variables are
different from each other. At each step of the algorithm both methods choose the
variable with the highest absolute correlation. The difference is that FS computes the
correlation using unshrunk coefficient estimates while the Lasso uses shrunk estimates.
Which approach works better? In general this is a complicated question. However,
Lemma 1 considers a simplified scenario, involving two signal predictors and one noise
variable, where specific conditions for success of each approach exist. For simplicity
we assume a linear model with two nonzero coefficients and no error term. We denote
by $\rho$ the correlation between the signal predictors and write $\phi_j$, $j \in \{1, 2\}$, for the
correlations between the signal predictors and the noise variable. We also assume that
the first signal variable is the one most highly correlated with the response, thus it is
the first variable selected by both the Lasso and FS. Simple calculations establish the
following result.

**Lemma 1** The first two variables selected by the Lasso are the “correct” signal pre-
dictors if and only if

\[
\{\phi_2 \geq \rho \phi_1 \text{ and } \rho - \phi_1 - \phi_2 > -1\} \text{ or } \{\phi_2 < \rho \phi_1 \text{ and } \rho + \phi_1 + \phi_2 > -1\}.
\] (3)

The first two variables selected by FS are the “correct” signal predictors if and only if

\[
\{\phi_2 \geq \rho \phi_1 \text{ and } \rho^2 - \rho \phi_1 + \phi_2 < 1\} \text{ or } \{\phi_2 < \rho \phi_1 \text{ and } \rho^2 + \rho \phi_1 - \phi_2 < 1\}.
\] (4)

The feasible combinations of the correlation parameters are given by

\[
\rho^2 + \phi_1^2 + \phi_2^2 - 2\rho \phi_1 \phi_2 \leq 1.
\] (5)

Figure 2 visualizes the results of Lemma 1 by providing the plots of the constraints
given by (3), (4), and (5) for three different values of $\phi_1$. The green solid lines
encompass the regions of feasible correlation combinations. The regions between the
blue dash dot curves correspond to the situations where FS will identify the correct
model. Alternatively, the regions above the red dashed curve represent the same
situations for the Lasso. Even in this simplified example it is clear that there are
many cases where FS succeeds and the Lasso fails, and vice versa.

Lemma 1 compares FS and the Lasso as the underlying population changes. How-
ever, even for two data sets generated from the same population, FS may provide
superior results on one data set while the Lasso works better for the other. Fig-
Figure 2: Plots showing regions where the Lasso and FS will identify the correct model for different correlation structures. Points above the red dashed lines correspond to the Lasso regions. Points between the blue dash dot lines correspond to FS. The green solid lines provide the regions of feasible correlation combinations.

Figure 3(a) illustrates this situation by plotting the squared $L_2$ distance ("L2 square") between the estimators and the true coefficient vector on 100 data sets. Each data set was simulated from a distribution with $p = 150$ variables, $n = 70$ observations, ten non-zero coefficients each with value 1, and pairwise correlations of 0.5 between the predictors. The elements of the design matrix were generated from a mean zero normal distribution, while the error terms were standard normal. The red dotted line corresponds to the L2 square values for FS sorted from lowest to highest, while the green dashed line represents the same quantities for the Lasso. Neither method universally dominates the other. The best 50% of the FS fits have lower L2 square than the corresponding fits for the Lasso. However, the worst 20% of the FS fits are considerably worse than those for the Lasso. Averaged over all 100 data sets the Lasso performs best, but for many of the data sets FS actually provides superior results. This plot represents a general trend among the data sets we examined: because of its lack of shrinkage in the coefficient estimates, FS demonstrates considerably more variability. When it works well its predictions are highly accurate, but at the other extreme it can produce very poor answers.

The black solid line in Figure 3(a) corresponds to the L2 square for a combined method that uses a separate validation data set, of size $n = 35$, to choose the best of the FS or Lasso fits. Interestingly, this approach has an average L2 square that is below the corresponding quantities for both the Lasso and FS. Figure 3(b) plots the FS and Lasso L2 square values for each of the 100 simulated data sets. The points
Lemma 1 and Figure 3 illustrate, not too surprisingly, that neither the Lasso nor FS universally dominate one another, and that there are situations where both will fail. However, the combined method mentioned above suggests that, for any given data set, it may be possible to automatically distinguish which method is performing best. In the next section we introduce our methodology which is capable of adaptively choosing the FS approach or the Lasso approach, or somewhere between the two. We show through extensive simulations and theoretical arguments that this adaptive procedure usually works no worse than the Lasso and FS and often outperforms both.

2.2 An Adaptive Shrinkage Methodology

Consider Figure 4, which illustrates just the first section of the plots in Figure 1. We have divided the figure into four regions. In region 1 the black solid absolute correlation is the largest, and hence both FS and the Lasso select this variable first. In region 2, the blue dash dot variable has the highest correlation and so it is selected second by the Lasso. Alternatively, in region 4 the red long dash variable has the highest correlation, hence it is selected second by FS. Our Forward-Lasso Adaptive Shrinkage (FLASH) methodology is based on the following observation. In Figure 4,
Figure 4: Region 1: All methods choose black solid variable first. Region 2: Blue dash dot variable has the maximum absolute correlation and is chosen by the Lasso. Region 3: Green dashed variable has the maximum absolute correlation but is not chosen by either the Lasso or FS. Region 4: Red long dash variable has the maximum absolute correlation and is chosen by FS.

Instead of stopping when the blue absolute correlation catches up to the black one (the shrunken Lasso rule) or when the black correlation reaches zero (the unshrunken FS rule), we propose to allow the data to determine how much shrinkage is appropriate at each step in the model selection path. FLASH automatically determines how far to drive the maximum correlation towards zero before selecting the next variable. In the example of Figure 4, FLASH could have a break point in region 2, in which case it would give the Blue Lasso model, or it could break in region 4, to give the Red FS model, or it may break in region 3 and choose an entirely new Green model.

Figure 5 illustrates a typical FLASH path. Figure 5(a) provides the initial path until the blue dash dot absolute correlation reaches the black solid one. The Lasso, FS and FLASH, all have identical paths up to this point. We have labeled this point $\delta_1 = 0$, while the point where the black correlation reaches zero is labeled $\delta_1 = 1$. Breaking at $\delta_1 = 0$ (maximum shrinkage) adds the blue variable, generating the Lasso solution, while breaking at $\delta_1 = 1$ (minimum shrinkage) adds the red long dash variable, generating the FS solution. FLASH treats $\delta_1$ as an adjustable parameter.
Figure 5: Sections of the FLASH correlation path. a) and b) represent the first section while c) and d) show the first two sections.

Figure 5(b) illustrates the FLASH path for one possible value, $\delta_1 = 1/2$. Here the first break point is midway between the Lasso and FS solutions and the green dash variable has the maximum absolute correlation, hence it is added to the model. Once the black and green variables have been selected, FLASH then constructs the path so as to drive both absolute correlations to reach zero simultaneously. In Figure 5(c) the first portion of the second section of the FLASH path has been plotted up to the point where the absolute correlation for one of the variables not currently in the model, in this case the red predictor, reaches the maximum. This point is labeled as $\delta_2 = 0$. The Lasso rule would break at this point and add the red variable to the model, while the FS rule would result in the path being continued to the point where the black and green correlations both reach zero, labeled $\delta_2 = 1$. However, FLASH again treats $\delta_2$ as an adjustable parameter. In Figure 5(d) we show the path for $\delta_2 = 1/2$. At this point the blue absolute correlation is the largest, hence the blue variable is added to the model. Next, the path would be constructed to drive the blue, green and black absolute correlations towards zero at such a rate as to reach zero simultaneously. This process continues, choosing new values for $\delta_3, \delta_4$ etc. until all correlations reach zero.
More formally, the path of all possible coefficient estimates from the FLASH methodology is produced using the algorithm below. Throughout the algorithm index set \( A \) represents the correlations that are being driven towards zero, and vector \( c_A \) contains the values of these correlations. We refer to this set and the corresponding correlations as "active". Note that the active absolute correlations are driven towards zero at rates that are proportional to their magnitudes. We use superscript \( l \) to denote each step of the algorithm, but for simplicity of the notation we omit the superscript wherever the meaning is clear without it.

1. Initialize \( \beta^1 = 0 \), \( A = \emptyset \) and \( l = 1 \).

2. Update the active set \( A \) by including the index of the (new) maximal absolute correlation. Let \( X_A \) be the matrix consisting of the columns of \( X \) associated with \( A \). Compute the \(|A|\)-dimensional direction vector \( h_A = (X_A^T X_A)^{-1} c_A \).

Let \( h \) be the \( p \)-dimensional vector with the components corresponding to \( A \) given by \( h_A \) and the remainder set to zero.

3. Compute \( \gamma_L \), the Lasso distance to travel in direction \( h \) until a new absolute correlation is maximal. We provide the formulas in the appendix, where we also show that \( \gamma_F \), the FS distance to travel in direction \( h \) until the active correlations reach zero, equals one. Define \( \gamma = \min(\gamma_L + \delta(1 - \gamma_L), 1) \) and let \( \beta^{l+1} = \beta^l + \gamma h \).

Set \( l \leftarrow l + 1 \).

4. Repeat steps 2 and 3 until all correlations are at zero.

### 2.3 Selection of Tuning Parameters

An important component of FLASH is the selection of the parameters \( \delta_k \). Clearly, treating each \( \delta_i \) as an independent tuning parameter is not feasible. Many model selection approaches could be utilized. In this paper we investigate two possible approaches. The first, "global FLASH", involves selecting a single value, \( \delta \), for all the step sizes, i.e. assuming a common level of shrinkage throughout the steps of the FLASH algorithm. Hence, \( \delta = 0 \) corresponds to the Lasso and \( \delta = 1 \) to FS. Using this approach we first choose a grid of \( \delta \)'s between 0 and 1 and then select the value giving the lowest residual sum of squares on a validation data set or, alternatively, the lowest cross-validated error. Global FLASH has the advantage of only needing to select one \( \delta \), which improves its computational efficiency.

The second approach, "block FLASH", allows for different values among the \( \delta_k \)'s. However, to make the problem computationally feasible, we constrain each \( \delta_k \) to be either zero or one. The version of block FLASH we focus on exclusively for the remainder of the paper (except in Section 3.3) involves selecting a single "break point" with \( \delta_j = 1 \) and setting all remaining \( \delta_k \)'s to zero. This has the effect of dividing FLASH into two stages. In the first stage a series of Lasso steps (i.e. \( \delta_k = 0 \)) are performed to select the initial variables. At the end of the first stage a Forward step
(i.e. $\delta_j = 1$) is performed which has the effect of removing the coefficient shrinkage on the currently selected variables. In the second stage further variables are selected by performing a series of Lasso steps. As with global FLASH, block FLASH has the advantage of only needing to select one tuning parameter, the break point. In Section 4 we provide simulation results for both versions of FLASH. In practice the two methods appear to perform similarly. However, as illustrated by Section 3, we are able to establish stronger theoretical properties for block FLASH than for global FLASH.

3 Theoretical Arguments

In this section we present some theoretical properties of FLASH. Section 3.1 demonstrates that FLASH can be formulated as the solution to a specific optimization problem, which draws connections between the Lasso and FLASH. Sections 3.2 and 3.3 present variable selection properties of FLASH and, in particular, conditions under which it can be shown to select the correct model. These conditions are weaker than the corresponding ones for the Lasso. Near optimal bounds on the $L_2$ error in the FLASH coefficient estimates are presented in Section 3.4. Finally, Section 3.5 provides a simple example where FLASH will outperform the Lasso and Forward Selection.

3.1 Optimization Criteria for FLASH

So far FLASH has been presented in terms of an algorithm. However, Theorem 1 below shows that, with a simple modification, FLASH constructs solutions to a weighted version of the standard Lasso optimization problem. The required modification is similar to the Lasso modification of the LARS algorithm, i.e. if at any point on the path a coefficient hits zero, then the corresponding variable is removed from the active set. A detailed description of this modification is given in the appendix. For the remainder of this paper when we refer to FLASH we mean the modified version.

**Theorem 1** Each point on the FLASH path is a solution to the weighted Lasso problem,

$$\min_{\hat{\beta}} \|Y - X\hat{\beta}\|^2_2 + \lambda \sum_{j=1}^{p} w_j |\hat{\beta}_j|,$$

where $w_1, w_2, \ldots, w_p$ is a set of non-negative weights. If the parameters $\delta_1, \delta_2, \ldots$ lie strictly below one, i.e. there are no FS steps, then the weights $w_1, w_2, \ldots, w_p$ are fixed for the entire FLASH path. Otherwise the weights are fixed between FS steps but reset at each $\delta_k = 1$.

Procedures using the solution to (6) have been previously suggested, for example in (Zou, 2006) and (Huang et al., 2006). The main potential practical weakness of these methods is that the weights need to be chosen ahead of time using some simple
method like the ordinary least squares. FLASH does not require choosing the weights in advance, they are essentially determined as the FLASH path is constructed. In practice we don’t use equation (6) to fit FLASH. The most significant implication of Theorem 1 is theoretical, because it draws a connection between FLASH and the Lasso and hence allows us to extend (with improvements) existing results to FLASH.

### 3.2 Variable Selection

For simplicity of exposition we study the variable selection properties of FLASH in the zero-noise case, but all the results that follow have probabilistic analogs in the general case. An advantage of this simplification is that all of our variable selection results are finite sample, i.e. stated for a fixed $n$. We define $K = \{ j : \beta_j \neq 0 \}$ and let $X_T$ denote the matrix consisting of the columns of $X$ associated with a given index set $T$. The notation we use for sub-vectors is analogous. We first consider the following *irrepresentable condition* on the design matrix, $X$,

$$||\Sigma_{K^cK}^{-1}\Sigma_{KK}^{-1}s_L||_\infty < 1.$$  \hfill (7)

Matrix $\Sigma_{K^cK}$, defined as $X_{K^c}^TX_{K^c}$, contains the correlations among the signal variables; matrix $\Sigma_{K^cK}$, defined as $X_{K^c}^TX_K$, contains the correlations between the signal and noise variables; and $s_L$ stands for the vector $\text{sgn}(\beta_K)$.

Zhao and Yu (2006), Zou (2006) and Meinshausen and Bühlmann (2006) show that the Lasso identifies the correct model with probability tending to one only if the irrepresentable condition holds. In the zero-noise setting for fixed $n$ and $p$ the results of Zhao and Yu also imply that if (7) holds, then the Lasso will “exactly recover” $\beta$, in other words it will pick the correct model for all small enough $\lambda$, and its coefficient path will converge to $\beta$ as $\lambda \downarrow 0$. If (7) is violated, counterexamples can be created where the Lasso fails to identify the correct model. Next we present a corresponding result for the simple two-stage version of the block FLASH method.

**Theorem 2** Suppose that the $qS$ variables that enter the model at the first stage are all signal variables. Define $s_{BF}$ analogously to $s_L$ except that the elements corresponding to variables that have entered the model are set to zero. Then, provided

$$||\Sigma_{K^cK}^{-1}\Sigma_{KK}^{-1}s_{BF}||_\infty < 1$$  \hfill (8)

holds, the block FLASH procedure will recover $\beta$ exactly.

Conditions (7) and (8) are very similar except that a fraction $q$ of the elements of $s_L$ are set to zero in $s_{BF}$. Intuitively, (8) seems less restrictive than (7), but because the two equations are so general, a direct comparison is not possible.

Instead we take a common approach of imposing bounds on the maximum absolute correlation between each pair predictors, under which (7) and (8) can be compared. In particular, the results of Zhao and Yu (2006) and Wainwright (2006) imply that
if the absolute values of the off-diagonal entries of the correlation matrix $X^T X$ are bounded above by $1/(2S - 1)$ then (7) will hold and hence the Lasso will recover $\beta$ exactly. The next result shows that (8) holds under weaker conditions.

**Theorem 3** Suppose that the $qS$ variables that enter the model at the first stage are all signal variables. If the off-diagonal entries of the correlation matrix $X^T X$ are bounded above in absolute value by $1/[((2-q)S)]$ inequality (8) will hold, and hence the block FLASH procedure will recover $\beta$ exactly.

A simple comparison of the block FLASH and Lasso conditions on the correlation matrix shows that the former condition is strictly less restrictive for $q > 1/S$. In fact, the correlation bound in Theorem 3 is not the tightest possible. Examination of the proof shows that the best possible block FLASH bound is strictly less restrictive than the Lasso bound for all positive $q$. More specifically, bound (15) can be used for small values of $q$. As $q$ approaches one, the block FLASH bound on the correlations grows to approximately twice that of the Lasso’s.

Theorems 2 and 3 both assume that the first few variables selected are “true” signal variables, and then show that in this situation block FLASH will recover the remaining variables under less restrictive conditions than for the Lasso. The intuition here is that for most regression problems there will be some fraction of the signal variables that are relatively easy to identify, while the remainder pose more difficulties. The block FLASH procedure utilizes the first group of signal variables in a more efficient fashion and hence is better able to identify the remaining predictors.

To mathematically quantify the intuition of easy and hard to identify signal variables we consider the situation where the non-zero elements of the vector $\beta$ split into two groups: $qS$ coefficients that are large in magnitude and $(1-q)S$ coefficients that are small in magnitude. Denote the smallest absolute value in the first group by $m_1$ and the largest absolute value in the second group by $M_2$. Theorem 4 below states that if the ratio of $m_1$ to $M_1$ is large enough and the absolute correlations in $X^T X$ are not too large, then block FLASH will correctly select both the “strong” and “weak” signal variables. In what follows $c_1$ and $c_2$ are positive functions of only $S$ and $q$ that can be bounded above by functions of just $q$. The formulas for them are given in display (16) of the appendix.

**Theorem 4** Suppose that $m_1/M_2 > c_1 S^{1/2}$. Let the first block of variables in the block FLASH procedure be identified using the Lasso solution corresponding to $\lambda = c_2 M_2$. If the absolute values of the off-diagonal entries of the correlation matrix are strictly below $1/\max\{2qS - 1, (2-q)S\}$, then this procedure will recover $\beta$ exactly.

The Lasso solution to which we refer corresponds to objective function (6) with the unit weights. The condition on the correlation matrix in Theorem 4 is optimized for $q \approx 2/3$. For smaller values of $q$ the bound is the same as that in Theorem 3.
3.3 Increasing the Number of Blocks

The previously discussed bounds can be improved by extending the two block structure we assumed for the non-zero coefficients of $\beta$ to the general case of $k^*$ blocks. Consider a partition of the set indexing the nonzero coefficients into subsets $K_j$ with $j \in \{1, ..., k^*\}$. Define

$$m_j = \min_{i \in K_j} |\beta_i| \quad M_j = \max_{i \in K_j} |\beta_i| \quad \text{and} \quad S_k = \sum_{i=1}^{k} |K_i|,$$

note that $S_{k^*} = S$ and set $S_0 = 0$ for convenience. Let the set $K_1$ correspond to the $|K_1|$ largest $|\beta_i|$'s, the set $K_2$ to the next $|K_2|$ largest $|\beta_i|$'s, and so on. In other words, our partition satisfies $m_i \geq M_j$ whenever $i < j$. Our block FLASH procedure now begins by performing the Lasso steps to get the first $|K_1|$ variables, followed by an FS step, then the Lasso steps to identify the next $|K_2|$ variables, followed by an FS step, and so on. As we did in the case $k^* = 2$, one can compute explicit formulas for the tuning parameters required to pick up each of the blocks of variables. However, the formulas become quite complicated, so we do not attempt to present them here.

**Theorem 5** Suppose that the absolute values of the off-diagonal entries of the correlation matrix $X^T X$ are bounded above by $1 / \max_j (2S_j - S_{j-1})$. Then, as long as the quantity $\min_j m_j/M_{j+1}$ is sufficiently large, the block FLASH procedure described above will exactly recover $\beta$.

Theorem 5 is an extension of Theorem 4 and, depending on the sizes of the $K_j$'s, potentially allows for a significant improvement in the bound on the correlation matrix. Corollary 1 illustrates the best case distribution of the $K_j$'s, maximizing the correlation bound.

**Corollary 1** If $|K_j| = 2|K_{j+1}|$ for each $j \geq 1$, then the bound on the absolute values of the off-diagonal entries of the correlation matrix needed in Theorem 5 can be taken as $(1 - 2^{-k^*})/S$.

Corollary 1 follows directly from Theorem 5. It shows that for moderate $k^*$ the bound on the correlations could be as high as $1/S$, compared to a bound of approximately half this size for the Lasso. However, even for other distributions of the $K_j$'s, the correlation bounds in this section are improvements over the Lasso bound of $1/(2S-1)$, for all values of $k^* > 1$.

3.4 Coefficient Estimation

Next we describe the coefficient estimation properties of block FLASH. To provide full generality we now consider the “noisy” setup of model (1), letting $\sigma^2$ be the variance of the error term $\epsilon$. Corollary 1 in Meinshausen and Yu (2009) establishes
that under the condition of “bounded maximal and minimal sparse eigenvalues” on
the correlation matrix $X^TX$ there exists a constant $M$, such that the bound

$$||\hat{\beta}_L - \beta||^2_2 \leq M\sigma^2S\log p$$  \hspace{1cm} (9)

holds with overwhelming probability for large $n$ and $p$. Here $\hat{\beta}_L$ is the Lasso solution corresponding to the tuning parameter $\lambda = \sigma\sqrt{\log p}$. For the precise conditions on the eigenvalues of $X^TX$ see Section 2.3 of Meinshausen and Yu. The two-stage block FLASH estimator $\hat{\beta}_{BF}$ satisfies an improved version of bound (9). As before, we assume that $qS$ variables enter the model at the first stage, and all of them are signal variables. Since we now assume noise in the model, $q$ should be treated as a random quantity taking values on $[0, 1]$.

**Theorem 6** Assume the conditions of Corollary 1 in Meinshausen and Yu (2009). Then, for the same constant $M$, the bound

$$||\hat{\beta}_{BF} - \beta||^2_2 \leq M\sigma^2S(1 - q)\log p$$  \hspace{1cm} (10)

holds with probability at least as high as the corresponding probability in the corollary. Here $\hat{\beta}_{BF}$ is the block FLASH estimator corresponding to $\lambda = \sigma\sqrt{\log p}$.

Note that the $\lambda$ in Theorem 6 refers to (6) because $\hat{\beta}_{BF}$ can be thought of as a solution to (6), where unit weights are used before the FS step, and then zero weights are assigned to the first block of variables. Theorem 1 in Meinhausen and Yu establishes a similar bound to (9) under slightly weaker assumptions on the sparse eigenvalues and can be adapted analogously to handle the block FLASH procedure. As in (10), the right hand side of the bound gets multiplied by $(1 - q)$, thus providing an improvement.

The intuition behind Theorem 6 is the same as that for Theorems 2 and 3: if there exist some signal variables that can be easily identified, block FLASH will give a strict improvement over the Lasso. This situation is formalized by considering two groups of nonzero coefficients, small and large, as in Theorem 4. In such a setting, provided there is a large enough separation between the two groups, the signal variables will be identified with very high probability.

### 3.5 An Illustrative Example

Here we examine a simple four variable example, involving two signal variables and no noise. In this situation we can graphically illustrate, for different correlation structures, situations were Lasso, FS or FLASH will choose the correct two variable model.

Figure 6 plots the absolute correlations for each of the four variables for the first part of the path. Each plot corresponds to different values of $\rho_{S_1,N_1}$ and $\rho_{S_1,N_2}$ where $\rho_{S_i,N_j}$ denotes the correlation between the $i$th signal and $j$th noise variable. The values for the other relevant parameters are fixed for all four plots at $\beta_1 = 2, \beta_2 = 1, \rho_{S_1,s_2} = 0.5, \rho_{S_2,N_1} = \rho_{S_2,N_2} = 0.8$. The signal and noise variables are represented...
using solid and dashed lines, respectively. In all four plots the black solid line is the highest at the beginning of the path, hence $S_1$ is selected first. Therefore plots where the solid lines cross each other first (the Lasso break point) represent situations where the Lasso will select the correct two variables. Alternatively, plots where the green solid line is highest when the black line reaches zero (the FS break point) represent situations where FS will select the correct two variables. Finally, any plots where, at some point, the green solid line is above all three other lines represent situations where FLASH will select the correct two variables. Here we can see that the Lasso works in Figures 6a) and c), while FS is correct for Figures 6a) and b). However, FLASH picks the correct model in all four situations, including Figure 6d) where both the Lasso and FS fail.
4 Simulation Results

In this section we present a detailed simulation study comparing FLASH to five natural competing approaches. We implemented both the global (FLASH$_G$) and block (FLASH$_B$) versions of our method discussed in Section 2.3. The tuning parameter $\delta$ in FLASH$_G$ was selected from a grid of five possible values, $\{0, .25, .5, .75, 1\}$. For the empirical studies in this paper we implemented a “relaxed” version of FLASH, which extends FLASH analogously to the way that the Relaxed Lasso extends the Lasso. We unshrink each solution located at a breakpoint of the FLASH path, connecting it via a path with the ordinary least squares solution on the corresponding set of variables. We do this as soon as the FLASH breakpoint is computed, in other words right after the third step of the algorithm. As with the Relaxed Lasso, the calculation of the corresponding relaxation direction comes at no computational cost, as it coincides with the current direction of the FLASH path. Indeed, if we continued the FLASH path along the current direction until all the active correlations simultaneously hit zero, we would arrive at exactly the ordinary least squares solution on the set of active variables.

We compared FLASH to VISA, Relaxed Lasso (Relaxo), the Adaptive Lasso (Adaptive), the path version of Forward Selection (Forward) and the Lasso. The Adaptive Lasso involves a preliminary step where the weights are typically chosen by performing a least squares fit to the data. This is not feasible for $p > n$, so we selected the weights using either the simple linear regression fits, as suggested in Huang et al. (2006), or a ridge regression fit, as suggested in Zou (2006). The ridged fits dominated so we only report results for the latter method here. In addition to the path version of FS we also implemented a version without a path and a version with shrinkage, where each Forward solution is connected to the zero vector via a linear path. The three methods performed similarly, so we only report the results for the first one.

Our simulated data consisted of five parameters which we varied: the number of variables ($p = 100$ or $p = 200$), the number of training observations ($n = 75$, $n = 105$ or $n = 150$), the correlations among the columns of the design matrix ($\rho = 0$ or $\rho = 0.5$), the number of non-zero regression coefficients ($S = 10$ or $S = 30$) and the standard deviation among the coefficients ($\sigma_\beta = 0.5$, $\sigma_\beta = 0.7$ or $\sigma_\beta = 1$). We tested most combinations of the parameters and report a representative sample of the results. The rows of the design matrix were generated from a mean zero normal distribution with a correlation matrix whose off-diagonal elements were equal to $\rho$. The error terms were sampled from the standard normal distribution while the regression coefficients were generated from a mean zero normal with variance $\sigma_\beta^2$. For each training data set we randomly sampled one-third of the observations to use as a validation set and selected the various tuning parameters for each method as those that gave the lowest mean squared error between the response and predictions on the validation data. In a practical application one might prefer to select the tuning parameters using a cross-validation approach but this was not feasible for our simulation study in which
we fitted each method 200 times for each simulation scenario. For each method and simulation we computed three statistics, averaged over 200 data sets: False Positive, the number of variables with zero coefficients incorrectly included in the final model; False Negative, the number of variables with non-zero coefficients left out of the model; and L2 square, the squared $L_2$ distance between the estimated coefficients and the truth. Table 1 provides the results.

The first four simulations correspond to $\rho = 0$ while the next four were generated using $\rho = 0.5$. The ninth simulation is a denser case with $S = 30$. Finally, the last four simulations represent harder problems with $\sigma_{\beta} = 0.7$ or 0.5, inducing lower signal to noise ratios. For the L2 square statistic we performed tests of statistical significance, comparing each method to the best FLASH approach. For each simulation we placed in bold the L2 square value for the best method and any other method that was not statistically worse at the 5% level of significance. For example, in the first simulation with 100 variables and 150 observations both versions of FLASH and FS were statistically indistinguishable from each other. However, in the third simulation with 100 variables and 75 observations FLASH$G$ was statistically superior to all other methods. Most of the standard errors for the L2 square statistic were relatively low, approximately 4% of the statistic’s value. However, as has been observed previously, we found that the Forward method often gave more variable estimates than the other approaches, with some standard errors as high as 8% of the statistic’s value.

None of the thirteen simulations contained a situation where one of the competing methods was statistically superior to FLASH, while in ten of the simulations FLASH was statistically superior to all other methods. In general FS performed best in the easiest scenarios with large $n$, zero correlation $\rho$ and higher signal, $\sigma_{\beta} = 1$. In particular, Forward performed very poorly in the denser $S = 30$ scenario, while this was the best situation for the Lasso. FLASH was still superior to both methods in this simulation setup. The Adaptive Lasso, VISA and Relaso all provided improvements over the Lasso, though the latter two methods generated the largest increase in performance. The two versions of FLASH performed at a similar level, though FLASH$_G$ seemed slightly better in the sparser cases while FLASH$_B$ was superior in the denser $S = 30$ situation. FLASH$_G$ also required less computational effort, because its path only needed to be computed once for each of the five potential values of $\delta$.

Overall, FS had low false positive but high false negative rates. In comparison to VISA and Relaso, FLASH$_G$ had the lowest false positive rates and similar or lower false negative rates. Alternatively, FLASH$_B$ had very low false negative rates and similar false positive rates.

5 Empirical Analysis

We examined FLASH on a data set containing salaries of professional baseball players (obtained from StatLib, Department of Statistics, CMU). For each player a number
<table>
<thead>
<tr>
<th>Simulation</th>
<th>Statistic</th>
<th>FLASH$_G$</th>
<th>FLASH$_B$</th>
<th>VISA</th>
<th>Relaxo</th>
<th>Adaptive</th>
<th>Forward</th>
<th>Lasso</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 150, p = 100$</td>
<td>False-Pos</td>
<td>1.92</td>
<td>3.32</td>
<td>3.23</td>
<td>3.7</td>
<td>9.9</td>
<td>1.11</td>
<td>18.68</td>
</tr>
<tr>
<td>$S = 10, \rho = 0$</td>
<td>False-Neg</td>
<td>2.12</td>
<td>1.89</td>
<td>2.26</td>
<td>2.26</td>
<td>1.84</td>
<td>2.33</td>
<td>1.27</td>
</tr>
<tr>
<td>$\sigma_\beta = 1$</td>
<td>L2-sq</td>
<td><strong>0.249</strong></td>
<td><strong>0.249</strong></td>
<td>0.292</td>
<td>0.308</td>
<td>0.342</td>
<td><strong>0.244</strong></td>
<td>0.436</td>
</tr>
<tr>
<td>$n = 150, p = 200$</td>
<td>False-Pos</td>
<td>1.99</td>
<td>3.91</td>
<td>3.53</td>
<td>3.87</td>
<td>12.61</td>
<td>1.07</td>
<td>21.18</td>
</tr>
<tr>
<td>$S = 10, \rho = 0$</td>
<td>False-Neg</td>
<td>2.32</td>
<td>2.09</td>
<td>2.44</td>
<td>2.45</td>
<td>2.44</td>
<td>2.48</td>
<td>1.64</td>
</tr>
<tr>
<td>$\sigma_\beta = 1$</td>
<td>L2-sq</td>
<td><strong>0.267</strong></td>
<td>0.286</td>
<td>0.353</td>
<td>0.366</td>
<td>0.524</td>
<td><strong>0.266</strong></td>
<td>0.606</td>
</tr>
<tr>
<td>$n = 75, p = 100$</td>
<td>False-Pos</td>
<td>2.65</td>
<td>6.17</td>
<td>4.88</td>
<td>5.1</td>
<td>10.39</td>
<td>1.71</td>
<td>15.41</td>
</tr>
<tr>
<td>$S = 10, \rho = 0$</td>
<td>False-Neg</td>
<td>3.3</td>
<td>2.9</td>
<td>3.38</td>
<td>3.4</td>
<td>3.08</td>
<td>3.79</td>
<td>2.42</td>
</tr>
<tr>
<td>$\sigma_\beta = 1$</td>
<td>L2-sq</td>
<td><strong>0.775</strong></td>
<td>0.848</td>
<td>0.996</td>
<td>1.021</td>
<td>1.228</td>
<td>0.929</td>
<td>1.285</td>
</tr>
<tr>
<td>$n = 75, p = 200$</td>
<td>False-Pos</td>
<td>3.73</td>
<td>7.24</td>
<td>6.46</td>
<td>6.84</td>
<td>12.89</td>
<td>1.71</td>
<td>18.54</td>
</tr>
<tr>
<td>$S = 10, \rho = 0$</td>
<td>False-Neg</td>
<td>3.83</td>
<td>3.4</td>
<td>4.06</td>
<td>4.04</td>
<td>3.81</td>
<td>4.57</td>
<td>3.04</td>
</tr>
<tr>
<td>$\sigma_\beta = 1$</td>
<td>L2-sq</td>
<td><strong>1.057</strong></td>
<td><strong>1.089</strong></td>
<td>1.477</td>
<td>1.496</td>
<td>1.999</td>
<td>1.365</td>
<td>1.934</td>
</tr>
<tr>
<td>$n = 150, p = 100$</td>
<td>False-Pos</td>
<td>3.13</td>
<td>4.79</td>
<td>6.33</td>
<td>6.53</td>
<td>10.41</td>
<td>1.32</td>
<td>19.66</td>
</tr>
<tr>
<td>$S = 10, \rho = 0.5$</td>
<td>False-Neg</td>
<td>2.59</td>
<td>2.33</td>
<td>2.48</td>
<td>2.45</td>
<td>2.21</td>
<td>3.02</td>
<td>1.62</td>
</tr>
<tr>
<td>$\sigma_\beta = 1$</td>
<td>L2-sq</td>
<td><strong>0.527</strong></td>
<td><strong>0.546</strong></td>
<td>0.629</td>
<td>0.656</td>
<td>0.661</td>
<td>0.581</td>
<td>0.797</td>
</tr>
<tr>
<td>$n = 150, p = 200$</td>
<td>False-Pos</td>
<td>3.35</td>
<td>6.35</td>
<td>7.06</td>
<td>7.33</td>
<td>11.82</td>
<td>1.27</td>
<td>21.72</td>
</tr>
<tr>
<td>$S = 10, \rho = 0.5$</td>
<td>False-Neg</td>
<td>3.12</td>
<td>2.88</td>
<td>3.06</td>
<td>3.11</td>
<td>3.01</td>
<td>3.57</td>
<td>2.23</td>
</tr>
<tr>
<td>$\sigma_\beta = 1$</td>
<td>L2-sq</td>
<td><strong>0.608</strong></td>
<td>0.673</td>
<td>0.752</td>
<td>0.785</td>
<td>0.872</td>
<td>0.655</td>
<td>1.029</td>
</tr>
<tr>
<td>$n = 75, p = 100$</td>
<td>False-Pos</td>
<td>5.12</td>
<td>8.31</td>
<td>7.23</td>
<td>7.44</td>
<td>11.27</td>
<td>2.42</td>
<td>16.2</td>
</tr>
<tr>
<td>$S = 10, \rho = 0.5$</td>
<td>False-Neg</td>
<td>3.95</td>
<td>3.38</td>
<td>3.79</td>
<td>3.88</td>
<td>3.53</td>
<td>4.82</td>
<td>2.99</td>
</tr>
<tr>
<td>$\sigma_\beta = 1$</td>
<td>L2-sq</td>
<td><strong>1.732</strong></td>
<td><strong>1.743</strong></td>
<td>1.84</td>
<td>1.901</td>
<td>2.088</td>
<td>2.38</td>
<td>2.199</td>
</tr>
<tr>
<td>$n = 75, p = 200$</td>
<td>False-Pos</td>
<td>5.82</td>
<td>9.62</td>
<td>8.8</td>
<td>8.77</td>
<td>12.91</td>
<td>2.37</td>
<td>18.28</td>
</tr>
<tr>
<td>$S = 10, \rho = 0.5$</td>
<td>False-Neg</td>
<td>5.14</td>
<td>4.45</td>
<td>5.04</td>
<td>5.12</td>
<td>4.9</td>
<td>6.25</td>
<td>4.34</td>
</tr>
<tr>
<td>$\sigma_\beta = 1$</td>
<td>L2-sq</td>
<td><strong>2.399</strong></td>
<td><strong>2.35</strong></td>
<td>2.648</td>
<td>2.7</td>
<td>2.851</td>
<td>3.094</td>
<td>2.934</td>
</tr>
<tr>
<td>$n = 75, p = 100$</td>
<td>False-Pos</td>
<td>10.6</td>
<td>13.73</td>
<td>11.34</td>
<td>12.09</td>
<td>15.62</td>
<td>4.39</td>
<td>17.16</td>
</tr>
<tr>
<td>$S = 30, \rho = 0$</td>
<td>False-Neg</td>
<td>14.23</td>
<td>12.1</td>
<td>14.12</td>
<td>13.89</td>
<td>12.91</td>
<td>21.7</td>
<td>11.95</td>
</tr>
<tr>
<td>$\sigma_\beta = 1$</td>
<td>L2-sq</td>
<td>10.559</td>
<td><strong>9.051</strong></td>
<td>10.749</td>
<td>10.743</td>
<td>11.132</td>
<td>19.792</td>
<td>11.316</td>
</tr>
<tr>
<td>$n = 150, p = 100$</td>
<td>False-Pos</td>
<td>3.54</td>
<td>4.72</td>
<td>6.09</td>
<td>6.19</td>
<td>10.52</td>
<td>1.84</td>
<td>17.86</td>
</tr>
<tr>
<td>$S = 10, \rho = 0.5$</td>
<td>False-Neg</td>
<td>3.73</td>
<td>3.54</td>
<td>3.51</td>
<td>3.56</td>
<td>3.34</td>
<td>4.24</td>
<td>2.46</td>
</tr>
<tr>
<td>$\sigma_\beta = 0.7$</td>
<td>L2-sq</td>
<td><strong>0.625</strong></td>
<td><strong>0.624</strong></td>
<td>0.692</td>
<td>0.705</td>
<td>0.724</td>
<td>0.707</td>
<td>0.877</td>
</tr>
<tr>
<td>$n = 150, p = 200$</td>
<td>False-Pos</td>
<td>4.06</td>
<td>6.21</td>
<td>7.11</td>
<td>7.48</td>
<td>11.69</td>
<td>1.68</td>
<td>21.46</td>
</tr>
<tr>
<td>$S = 10, \rho = 0.5$</td>
<td>False-Neg</td>
<td>4.05</td>
<td>3.81</td>
<td>3.87</td>
<td>3.93</td>
<td>3.7</td>
<td>4.68</td>
<td>3</td>
</tr>
<tr>
<td>$\sigma_\beta = 0.7$</td>
<td>L2-sq</td>
<td><strong>0.686</strong></td>
<td>0.731</td>
<td>0.788</td>
<td>0.794</td>
<td>0.799</td>
<td>0.77</td>
<td>0.922</td>
</tr>
<tr>
<td>$n = 150, p = 100$</td>
<td>False-Pos</td>
<td>3.81</td>
<td>4.88</td>
<td>6.11</td>
<td>5.93</td>
<td>9.65</td>
<td>1.99</td>
<td>15.78</td>
</tr>
<tr>
<td>$S = 10, \rho = 0.5$</td>
<td>False-Neg</td>
<td>4.74</td>
<td>4.49</td>
<td>4.46</td>
<td>4.51</td>
<td>4.16</td>
<td>5.48</td>
<td>3.42</td>
</tr>
<tr>
<td>$\sigma_\beta = 0.5$</td>
<td>L2-sq</td>
<td><strong>0.559</strong></td>
<td><strong>0.545</strong></td>
<td>0.576</td>
<td>0.584</td>
<td>0.596</td>
<td>0.683</td>
<td>0.633</td>
</tr>
<tr>
<td>$n = 150, p = 200$</td>
<td>False-Pos</td>
<td>3.69</td>
<td>5.25</td>
<td>5.76</td>
<td>6.13</td>
<td>10.11</td>
<td>1.54</td>
<td>17.73</td>
</tr>
<tr>
<td>$S = 10, \rho = 0.5$</td>
<td>False-Neg</td>
<td>5.38</td>
<td>5.08</td>
<td>5.29</td>
<td>5.32</td>
<td>4.88</td>
<td>6.03</td>
<td>4.24</td>
</tr>
<tr>
<td>$\sigma_\beta = 0.5$</td>
<td>L2-sq</td>
<td><strong>0.664</strong></td>
<td><strong>0.662</strong></td>
<td><strong>0.696</strong></td>
<td>0.709</td>
<td>0.715</td>
<td>0.761</td>
<td>0.769</td>
</tr>
</tbody>
</table>

Table 1: Simulation results for each method.
of statistics were recorded, such as career runs batted in, walks, hits, at bats, etc. We then used these variables to predict salaries. After including all possible interaction terms, the data set contained $n = 263$ observations and $p = 153$ predictors. This represented a somewhat difficult regression problem, because of the large number of predictors relative to the number of observations.

We tested three possible competitors to FLASH, namely Lasso, FS and the Relaxed Lasso. For each of the four methods ten-fold cross-validation was used to compute the root mean squared error (RMSE) in prediction accuracy at various points of the coefficient path. The final results are illustrated in Figure 7(a). The open green circles represent the Lasso error rates at the end of each step of the LARS algorithm. Alternatively, the green solid dots show the least squares fits for the models selected by the Lasso. The green dashed line illustrates the Relaxed Lasso fit as the coefficient shrinkage is reduced from the Lasso estimate (maximum shrinkage) to the least squares fit (no shrinkage). The red dash-dot line corresponds to FS. This is a step function because no shrinkage is used to estimate the coefficients so the fit remains constant within each step. Finally, the black solid line represents the FLASH fit with $\delta = 0.25$. Again, within each step we “relax” the initial FLASH estimate to the corresponding least-squares solution, as described in the first paragraph of Section 4.

For roughly the first fifteen steps FS has strictly lower cross-validated error rates.
Table 2: Coefficients and confidence intervals for the eight statistically significant variables selected by FLASH when fitting the baseball data. The variables are listed in the order that they entered. Significance records the fraction of times the variable was excluded from the bootstrap models.

<table>
<thead>
<tr>
<th>Order</th>
<th>Variable</th>
<th>Coefficient</th>
<th>Confidence Interval</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Career RBI * Hits</td>
<td>0.004</td>
<td>(0.001, 0.007)</td>
<td>0.00</td>
</tr>
<tr>
<td>2.</td>
<td>Walks * Hits</td>
<td>0.072</td>
<td>(0.030, 0.110)</td>
<td>0.00</td>
</tr>
<tr>
<td>3.</td>
<td>Career Walks * Career HmRun</td>
<td>−0.001</td>
<td>(−0.004, 0.000)</td>
<td>0.00</td>
</tr>
<tr>
<td>4.</td>
<td>Assists * HmRun</td>
<td>−0.044</td>
<td>(−0.048, −0.019)</td>
<td>0.01</td>
</tr>
<tr>
<td>5.</td>
<td>At Bat</td>
<td>−1.278</td>
<td>(−1.487, −0.786)</td>
<td>0.01</td>
</tr>
<tr>
<td>6.</td>
<td>Career At Bat * Years</td>
<td>−0.012</td>
<td>(−0.024, −0.006)</td>
<td>0.02</td>
</tr>
<tr>
<td>7.</td>
<td>Assists</td>
<td>1.200</td>
<td>(0.152, 2.328)</td>
<td>0.03</td>
</tr>
<tr>
<td>8.</td>
<td>Career Runs</td>
<td>1.647</td>
<td>(0.560, 2.579)</td>
<td>0.03</td>
</tr>
</tbody>
</table>

than those of Lasso or the relaxed Lasso. Correspondingly, FLASH is superior to the latter two methods. However, from step 8 onwards, FLASH also outperforms FS. In fact from step 15 onwards, where FS begins to significantly deteriorate, FLASH continues to improve and eventually achieves the lowest error rate of all three methods at approximately step 20. Figure 7(b) plots the cross-validated error paths out to 80 steps. The relaxed Lasso achieves its optimal results at approximately step 50, which corresponds to a 34 variable model. Not only is the optimal error rate higher than that for FLASH but the corresponding model contains twice as many variables as the model selected by FLASH, which only had 17 variables. This effect is not due to over-shrinkage in the final coefficient estimates, because the relaxed Lasso is capable of adjusting the level of shrinkage. Our simulation results pointed to a similar phenomenon and we have noticed in other real data sets that FLASH tends to select sparser models, suggesting FLASH may have an advantage in terms of inference in addition to prediction accuracy.

Table 2 provides the eight statistically significant variables, along with their coefficients and confidence intervals, from the 17 variable FLASH model selected at step 20. To test the significance of the variables we generated 100 bootstrap data sets from the baseball data and fitted FLASH to each one of them. The confidence intervals were generated using the 2.5 and 97.5 percentiles of the FLASH estimates on the bootstrap data. Number of at bats in the previous season along with assists and career runs enter as main effects. In addition career RBI’s, hits, walks, career walks, career home runs, assists, home runs, career at bats and years as a player all enter as interaction effects.

6 Discussion

Both FS and the Lasso iteratively build models by including the variable with the highest correlation to the current estimate of the residual vector. The only difference
is in how much shrinkage they use to estimate the residuals. FS applies no shrinkage, while the Lasso often uses a great deal of shrinkage. Generally, the FS approach works best on relatively easy problems with strong signal, while the Lasso provides superior results on harder problems. However, as we have demonstrated, for any given data set there is no particular reason that either of the extremes of shrinkage will produce the best solution. FLASH allows the data to dictate the optimal level of shrinkage in the model selection stage. This is quite different from approaches such as Relaxo that, while they adjust the level of shrinkage after the model has been selected, do not adjust the shrinkage when choosing the sequence of models to consider. As a result, FLASH often produces sparser models with superior predictive accuracy.

Computational efficiency is always important for high-dimensional problems. The standard FLASH algorithm is very similar to LARS and hence involves a relatively small computational expense. However, the fact that FLASH can be formulated as a weighted Lasso optimization problem (Theorem 1) introduces the possibility of even more efficient fitting procedures. One obvious example of such a procedure can be found in the recent work on pathwise coordinate descent algorithms (Friedman et al., 2007), which appear to be computationally superior to even LARS. These algorithms can easily be adapted to weighted Lasso problems, hence it should be possible to apply them to FLASH. While it does not seem easy to compute the weights for (6) in advance, it is conceptually simple to iteratively calculate each weight as the FLASH path is constructed.

A Step 3 of the FLASH algorithm

Let $c_{is}$ be one of the active correlations with the maximum absolute value. Then, as with LARS, the first time a non-active absolute correlation reaches the “active” maximum corresponds to the step size of

$$
\gamma_L = \min_{j \in A^c} \left\{ \frac{c_{is} - c_j}{(x_{is} - x_j)^T X h}, \frac{c_{is} + c_j}{(x_{is} + x_j)^T X h} \right\}.
$$

Along the direction $h$ all active correlations reach zero at the same time. Hence, the FS step size is given by

$$
\gamma_F = \frac{c_{is}}{x_{is}^T X h} = \frac{c_{is}}{x_{is}^T X_A (X_A^T X_A)^{-1} c_A} = 1.
$$

B Weighted Lasso modification

The basic FLASH algorithm described in Section 2.2 shares the following property with the basic LARS algorithm: once a variable enters the model, it does not leave. Recall that the Lasso solution path can be obtained from the modified LARS algorithm, where if a coefficient hits zero, the corresponding variable is removed from the
active set, and hence the model as well. When a variable is removed, the corresponding absolute correlation goes below the value it would be at if it remained active. The variable rejoins the model if its absolute correlation reaches the value it would be at if it stayed in the model. We provide a similar modification to the FLASH algorithm. Theorem 1 implies that this modification is well defined.

**Definition 1 (Weighted Lasso modification)** When a coefficient hits zero, the corresponding variable is removed from the active set. The variable is added back to the active set once the corresponding absolute correlation reaches the value it would currently be at had it remained active. Also, while the variable is out of the active set, it is ignored in the calculation of the maximum absolute correlation in step 2 of the FLASH algorithm.

In LARS it is easy to keep track of what the absolute correlation value would be if the removed variable remained active: it is just the value of the maximum absolute correlation. In FLASH this value is also easy to keep track of, because all pairwise ratios among the active absolute correlations stay fixed throughout the algorithm.

**C Proof of Theorem 1**

First consider a point on the FLASH path before which there have been no zero crossings. As before, let $c_i$ denote the correlation between predictor $X_i$ and the current residual vector, and let $\mathcal{A}$ be the current active set. Also define $c_{\text{max}} = \max_{i \in \mathcal{A}} |c_i|$ and $c_{\text{min}} = \min_{i \in \mathcal{A}} |c_i|$. Note that, by the construction of the path, each new nonzero coefficient $\hat{\beta}_i$ comes in with the same sign as the current correlation $c_i$ (the proof of this fact is the same as for the original LARS algorithm). Consequently, the FLASH estimator $\hat{\beta}$ satisfies the following set of conditions,

\[
X_i^T (Y - X \hat{\beta}) = |c_i| \text{sgn}(\hat{\beta}_i), \quad i \in \mathcal{A},
\]

\[
|X_j^T (Y - X \hat{\beta})| \leq c_{\text{max}}, \quad i \in \mathcal{A}^c.
\]

But these are exactly the KKT conditions for the weighted Lasso problem (6) with, for example, $\lambda = 2c_{\text{min}}$, $w_i = |c_i|/c_{\text{min}}$ for $i \in \mathcal{A}$, and $w_j \geq c_{\text{max}}/c_{\text{min}}$ for $j \in \mathcal{A}^c$. Note that the weights satisfying these conditions can be defined sequentially as new variables are added to the model. Each new weight is greater than the weights already defined and the weights for the active variables are proportional to the corresponding absolute correlations. Also note that a forward step resets all the weights for the active variables to zero.

Zero crossings are handled by the weighted lasso modification exactly the same way the corresponding weighted Lasso path algorithm would handle it (essentially, this is the reason for the modification). To illustrate this fact suppose, for concreteness, that the very first variable added stays in the model throughout, and suppose that $X_1$ leaves
the model after its coefficient hits zero. Then, according to the modification, \( X_i \) will be added back to the model when and only when \(|c_i|\) catches up to the level \( w_i c_{\text{min}} \), where \( w_i \) is fixed and \( c_{\text{min}} \) is decreasing along the FLASH path. This is exactly the same as what would happen on the corresponding weighted Lasso path.

\[ \text{D Proof of Theorem 2} \]

Let \( K_1 \) denote the set indexing the variables in the first block. Consider a FLASH tuning parameter \( \lambda \) small enough to ensure that the corresponding block FLASH estimator, call it \( \hat{\beta} \), is past the forward step when viewed as a point on the path. Recall that \( \hat{\beta} \) solves the weighted Lasso problem

\[
\hat{\beta} = \arg \min_{\beta} \|Y - X\beta\|_2^2 + \sum_k \lambda_k |\tilde{\beta}_k|,
\]

where the weights are given by \( \lambda_k = \lambda I_{\{k \in K_1\}} \). KKT conditions for the weighted Lasso problem imply that if \( \hat{\beta} = \beta + h \) and \( h \) satisfies

\[
2X_i^T Xh = -\lambda_i \text{sgn}(\hat{\beta}_i), \quad \hat{\beta}_i \neq 0, \tag{12}
\]

\[
2|X_j^T Xh| < \lambda_j, \quad \hat{\beta}_j = 0, \tag{13}
\]

then \( \hat{\beta} \) is a block FLASH solution corresponding to the tuning parameter \( \lambda \).

Recall the definition of \( s_{\text{BF}} \) from Section 3.2 and introduce the estimator \( \hat{\beta} \) as \( \beta + h \) with \( 2h_K = -\lambda(X_K^T X_K)^{-1}s_{\text{BF}} \) and \( h_{K^c} = 0 \). It is only left to check that \( h \) satisfies conditions (12) and (13). Note that for small enough \( \lambda \) vectors \( \hat{\beta} \) and \( \beta \) have exactly the same sign pattern, and hence condition (12) is satisfied by the definition of \( h \). Condition (13) becomes

\[
\|X_{K^c}^T X_K (X_K^T X_K)^{-1}s_{\text{BF}}\|_\infty < 1,
\]

which is precisely assumption (8) of our theorem.

\[ \text{E Proof of Theorem 3} \]

Write \( \|A\| \) for the operator norm of a matrix \( A \), let \( \bar{q} \) stand for \( 1 - q \), and let all the absolute values of the off-diagonal elements of \( X^T X \) be bounded above by \( B \). Our goal is to find the largest \( B \) for which inequality (8) is automatically satisfied. Note that \( \Sigma_{K^c K}^{-1}s_{\text{BF}} = \Sigma_{K^c K}^{-1}E s_{\text{BF}} \), where \( E = \text{diag}(|s_{\text{BF}}|) \) and the \( |\cdot| \) operation is understood componentwise. It follows that

\[
\|\Sigma_{K^c K}^{-1}s_{\text{BF}}\|_\infty \leq BS\bar{q}^{1/2}\|\Sigma_{K^c K}^{-1}E\|.
\]
Define $A = I - \Sigma_{KK}$ and note that if inequality $\|A\| < 1$ holds, then $\Sigma_{KK}^{-1} = \sum_{j=0}^{\infty} A^j$ and $\|\Sigma_{KK}^{-1} E\| \leq 1 + \|AE\|/(1 - \|A\|)$. Arguing exactly as Zhao and Yu (2006) in their display (67), but without explicitly specifying the correlation bound, we get $\|A\| \leq B(S - 1)$ and $\|EA\| \leq B(qS - 1)$. Consequently, if we require $B < 1/(S - 1)$, the needed bound $\|A\| < 1$ holds, and the preceding argument goes through. Conclude that

$$\|\Sigma_{KK}\Sigma_{KK}^{-1} s_{BF}\|_\infty \leq BSq^{1/2}[1 - BqS]/[1 - B(S - 1)].$$

(14)

Now we need to equate the right hand side to one and solve for $B$. We don’t have a closed form for the solution, but a direct calculation shows that the right-hand side of (14) is strictly below one for $B = 1/[(2 - q)S]$. This completes the proof of our theorem.

To obtain a better bound for very small values of $q$ note that the right-hand side of (14) is strictly below one for each $B$ below

$$[(1 + \bar{q}^{1/2})S - 1]^{-1}, \quad \text{with } \bar{q} = 1 - q,$$

(15)

which can be seen by replacing the $q$ in (14) with zero and solving for $B$.

F Proof of Theorem 4

Let $K_1$ and $K_2$ index the “large” and the “small” coefficients, respectively. To simplify the notation, replace expressions like $K_1, K_2, K_1^c$ in various subscripts by $1, 2, 1^c$, etc. Define $s_1 = \text{sgn}(\beta_1)$ and introduce $\hat{\beta}$ by setting $\hat{\beta}_{1c} = 0$ and writing $\hat{\beta}_1$ as $\beta_1 + h_1$ with $h_1 = \Sigma_{i1}^{-1}(-\lambda s_1/2 + \Sigma_{i2}\beta_2)$. We want to show that $\hat{\beta}$ solves the Lasso problem for some tuning parameter $\lambda$, because this would mean we are able to pick up exactly the variables in $K_1$ at the first stage of our block FLASH procedure. Theorem 3 then guarantees that our procedure will exactly recover $\beta$ as long as the correlation bound $B < 1/[(2 - q)S]$ is satisfied. Note that even in the best case scenario $M_2 = 0$ we cannot get around the usual Lasso condition $B < 1/[2qS - 1]$ if we want to ensure the correct first block of variables is selected. Hence we will require $B < B^*$ with $B^* = 1/\text{max}\{2qS - 1, (2 - q)S\}$.

We will check if the Lasso KKT conditions hold for $\hat{\beta}$. Note that if $\|h_1\|_\infty < m_1$, then $\hat{\beta}_1$ and $\beta_1$ have the same sign pattern. If this is the case, then the KKT equality conditions become $X_1^T(-X_1h_1 + X_2\beta_2) = \lambda s_1/2$ and are satisfied automatically. The KKT inequality conditions become $\|X_1(-X_1h_1 + X_2\beta_2)\|_\infty < \lambda/2$. All that is left to do is write a set of requirements on $m_1, M_2$ and $\lambda$ under which the last inequality holds together with $\|h_1\|_\infty < m_1$. Linear algebra arguments along the lines of those in the previous section show that the desired inequalities follow from

$$[\lambda q SB^* + M_2 SB^*(1 + [1 - 2q]SB^* + B^*)]/[1 - q SB^* + B^*] \leq \lambda/2 \quad \text{and}$$

$$(qS)^{1/2}[\lambda/2 + M_2 \bar{q}SB^*]/[1 - q SB^* + B^*] \leq m_1.$$
Rearranging the terms yields that the above conditions are satisfied if \( \frac{m_1}{M_2} > c_1 S^{1/2} \) and \( \lambda = c_2 M_2 \), where

\[
c_1 = q^{1/2} \left[ \frac{c_2}{2 + SB^* q} \right] / \left[ 1 - qSB^* + B^* \right] \quad \text{and} \quad c_2 = 2SB^* \left( 1 + \left| 1 - 2qSB^* + B^* \right| / \left[ 1 - 2qSB^* + B^* \right] \right) \quad \text{with} \quad B^* = 1 / \max \{ 2qS - 1, (2 - q)S \} \quad \text{and} \quad \bar{q} = 1 - q.
\]

### G Proof of Theorem 5

The fact that the separation between groups is arbitrarily large allows as to act essentially as if all the coefficients in the groups \( \{ K_j, j > i \} \) were exactly zero when we derive the upper bound on the correlations sufficient to identify the coefficients in group \( K_i \). Consequently, to identify the coefficients in \( K_1 \) we need the upper bound \( B \) to be strictly below \( 1 / [2S_1 - 1] \), as prescribed by the usual Lasso bound; to identify the coefficients in \( K_2 \) having identified the ones in \( K_1 \) we need \( B \) to be strictly below \( 1 / [2S_2 - S_1] \), as prescribed by the bound from our Theorem 3; and so on.

### H Proof of Theorem 6

We will modify the proof of Theorem 1 in Meinshausen and Yu (2009) (“MY” from here on) to handle the block FLASH procedure. To simplify the notation we will use the symbol \( \hat{\beta} \) to refer to both the Lasso estimator in MY and our block FLASH estimator. It will be clear from the context which estimator we are considering.

The quantity of interest is bounded by the sum of two terms that represent the variance and the squared bias of the estimation, respectively:

\[
\frac{1}{2} \| \hat{\beta} - \beta \|_2^2 \leq \| \hat{\beta} - \hat{\beta}_0 \|_2^2 + \| \hat{\beta}_0 - \beta \|_2^2.
\]

Here \( \hat{\beta}_0 \) is the corresponding estimator under the absence of noise, i.e. when the error term \( \epsilon \) is removed from the linear model (1). MY show that the variance of the estimation is bounded above by \( b_1 \sigma^2 S \log p \) with probability tending to one, and the bias is bounded above by \( b_2 \sigma^2 S \log p \). Constants \( b_1 \) and \( b_2 \) satisfy \( b_2 / b_1 > 150 \), so the bias term clearly dominates. In fact, it follows from the proof that \( b_1 \) can be replaced by a quantity that goes to zero as \( n \) and \( p \) tend to infinity (see, for example, the comment below inequality (40) in MY). We will show that the variance of our \( \hat{\beta} \) satisfies the same bound as in MY, while the bias satisfies an improved bound with \( b_2 \) replaced by \( b_2 (1 - q) \), and this is exactly what we need to prove our theorem.
H.1 Bias

All we need to show is that there is an extra $(1 - q)^{1/2}$ factor on the right hand side of the inequality established in Lemma 1 in MY. This lemma bounds the $L_2$ norm of the vector $\hat{\beta}_0 - \beta$, denoted by $\gamma$. The bound is based on the inequality

$$\|X\gamma\|_2^2 \leq \lambda S^{1/2}\|\gamma\|_2,$$  \hspace{2cm} (17)

combined with a quadratic lower bound on $\|X\gamma\|_2^2$. The above display is just inequality (21) in MY written in our notation. The aforementioned quadratic lower bound is based on the eigenvalue properties of $X^TX$ and holds for our block FLASH estimator as well. We will improve the right hand side of inequality (17) by introducing an extra $(1 - q)^{1/2}$ factor.

Inequality (17) for the Lasso case comes from the fact that $\gamma$ minimizes the function $f(\zeta) = \|X\zeta\|_2^2 + \lambda \sum_{k \in K^c} |\zeta_k| + \lambda \sum_{k \in K} (|\beta_k + \zeta_k| - |\beta_k|)$. Indeed, it follows from $f(\gamma) \leq 0$ and an application of the Cauchy-Schwartz inequality that

$$\|X\gamma\|_2^2 \leq \lambda \sum_{k \in K} |\gamma_k| \leq \lambda (S \sum_{k \in K} |\gamma_k|^2)^{1/2} \leq \lambda S^{1/2}\|\gamma\|_2.$$  \hspace{2cm} (18)

The argument for our $\hat{\beta}$ is almost identical, but the penalty weights $\lambda$ in the definition of $f(\gamma)$ now depend on $k$ with $\lambda_k = \lambda 1_{\{k \in K^c\}}$. As a result, inequalities (18) become

$$\|X\gamma\|_2^2 \leq \sum_{k \in K} \lambda_k |\gamma_k| \leq \lambda (|K_2| \sum_{k \in K_2} |\gamma_k|^2)^{1/2} \leq \lambda |K_2|^{1/2}\|\gamma\|_2,$$

which is exactly what we need because $|K_2| = (1 - q)S$.

H.2 Variance

For each $\xi$ in $[0, 1]$ let $A_\xi$ denote the set of active variables corresponding to the estimator under consideration when the error term $\epsilon$ in the linear model (1) is replaced by $\xi \epsilon$. Lemmas 3 and 4 of MY establish the following upper bound for the variance with probability tending to one,

$$\|\hat{\beta} - \hat{\beta}_0\|_2^2 \leq 2\sigma^2 \max_{\xi \in [0, 1]} |A_\xi| \log p.$$

The lemmas and the bound on the variance actually hold in the more general weighted Lasso set up, thus they hold for our block FLASH estimator as well.

Lemma 5 in MY establishes $\max_{\xi \in [0, 1]} |A_\xi| \leq b_3 S$ for the Lasso estimator with probability tending to one. For the purposes of MY’s Corollary 1 and our Theorem (6) $b_3$ is some large positive constant. We will now argue that the same probability bound is valid for our estimator. At the end of the proof MY establish an inequality that
can be written as
\[ \max_{\xi \in [0,1]} |A_\xi| \leq b_4 \text{Bias}^2 / (\sigma^2 \log p) \]
for a certain constant \( b_4 \). An examination of the proof reveals that the above inequality also holds for the more general weighted Lasso setup if \( |A_\xi| \) is replaced by the corresponding maximum of the number of nonzero equality constraints in the KKT conditions. In the Lasso case this number simply equals \( |A_\xi| \). In the block FLASH scenario this number is smaller, because \( qS \) of the equality constraints are exactly zero. We only need to handle the case \( \max_{\xi \in [0,1]} |A_\xi| > S \), hence our estimator satisfies
\[ (1 - q) \max_{\xi \in [0,1]} |A_\xi| \leq b_4 \text{Bias}^2 / (\sigma^2 \log p) \]
Using our results for the bias, we conclude that the probability bound on the variance established by MY for the Lasso is valid for our estimator as well.

References


