VIF Regression: A Fast Regression Algorithm for Large Data

Dongyu LIN, Dean P. FOSTER, and Lyle H. UNGAR

We propose a fast and accurate algorithm, *VIF regression*, for doing feature selection in large regression problems. VIF regression is extremely fast; it uses a one-pass search over the predictors and a computationally efficient method of testing each potential predictor for addition to the model. VIF regression provably avoids model overfitting, controlling the marginal false discovery rate. Numerical results show that it is much faster than any other published algorithm for regression with feature selection and is as accurate as the best of the slower algorithms.

KEY WORDS: Marginal False Discovery Rate; Model selection; Stepwise regression; Variable selection.

1. INTRODUCTION

Datasets from such areas as genetic sequences, text mining the Web, image processing, and sensor networks can now easily contain millions of observations and hundreds of thousands of features. Even a medium-sized dataset can create a huge number of potential variables if interactions are considered. The problem of variable selection or feature selection, which aims to select the most predictive of an enormous number of candidate features, plays an increasingly important role in modern research (Guyon and Elisseeff 2003).

The specific problem that we consider here is how to improve the speed of variable selection algorithms for linear regression models of very large-scale data. Linear regression models are widely used for building models for large problems; their simplicity makes them fast and easy to evaluate.

The statistical embodiment of variable selection that we consider here is a classical normal linear model,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1}$$

with n observations $\mathbf{y} = (y_1, \ldots, y_n)'$ and p predictors $\mathbf{x}_1, \ldots, \mathbf{x}_p$, $p \gg n$, where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$ is an $n \times p$ design matrix of features, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ is the vector of coefficient parameters, and error $\boldsymbol{\varepsilon} \sim \mathrm{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$.

The number of the features in the dataset is often much larger than the number of the observations. In these cases, we need to either regularize the coefficient parameters $\boldsymbol{\beta}$ in (1) or select a subset of variables that can provide a jointly predictive model, assuming that only a subset of k of the p predictors $\{\mathbf{x}_j\}_{j=1}^p$ in (1) has nonzero coefficients (Miller 2002). In this article we present a fast algorithm for searching for such a low-dimensional model.

Our variance inflation factor (VIF) regression algorithm has a computation complexity, $O(pn)$, under the sparsity assumption that $k \ll p$. This speed enables the VIF algorithm to handle larger datasets than many competitors, as illustrated in Figure 1. The VIF regression algorithm also guarantees good control of the marginal false discovery rate (mFDR) (Foster and Stine 2008) with no overfitting, and thus provides accurate predictions. Figure 2 shows the out-of-sample performance of VIF and four competing algorithms. VIF regression is more accurate than its fastest competitor, GPS (Friedman 2008), and is of comparable accuracy to its slow but accurate competitors, such as stepwise regression.

1.1 Related Work

Variable selection algorithms are generally designed to seek an estimate of $\boldsymbol{\beta}$ that minimizes the l_q penalized sum of squared errors

$$\arg\min_{\boldsymbol{\beta}} \left\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_q \|\boldsymbol{\beta}\|_{l_q} \right\}, \tag{2}$$

where $\|\boldsymbol{\beta}\|_{l_q} = (\sum_{i=1}^p |\boldsymbol{\beta}_i|^q)^{1/q}$ for $q > 0$ and $\|\boldsymbol{\beta}\|_{l_0} = \sum_{i=1}^p I_{\{\beta_i \neq 0\}}$.

The aforementioned problem of selecting a subset of variables corresponds to using an l_0 norm in (2). This problem is NP hard (Natarajan 1995), yet its solution can be greedily approximated by *stepwise regression*, a standard statistical tool. Stepwise regression works well for moderate-sized datasets, but has relatively high computational complexity, $O(np^2q^2)$. It can become very slow when n is large, because $o(n/\log n)$ variables can enter the model without overfitting (Breiman and Freedman 1983; Greenshtein and Ritov 2004). Zhang (2009) developed a new optimization algorithm, FoBa, which also addresses the l_0 problem and provides a theoretical bound on its accuracy. But FoBa is extremely slow, as shown in our experiments; also, unlike VIF regression, it requires cross-validation to determine the sparsity of the model.

A rich literature has been developed in recent years solving (2) using an l_1 norm penalty. Exact solutions can be found efficiently because of the convexity of the l_1 problem, for example, Lasso/LARS (Efron et al. 2004) and the Dantzig Selector (Candes and Tao 2007). These l_1 methods have several limitations, however. First, cross-validation is needed to determine the penalty λ_1; this is time-consuming and is not realizable in the setting where predictors are generated dynamically.

Dongyu Lin is Postdoctoral Fellow, Department of Biostatistics and Epidemiology, University of Pennsylvania School of Medicine, Philadelphia, PA 19104 (E-mail: *dongyu@wharton.upenn.edu*). Dean P. Foster is Marie and Joseph Melone Professor, Department of Statistics, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104 (E-mail: *dean@foster.net*). Lyle H. Ungar is Associate Professor, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 (E-mail: *ungar@cis.upenn.edu*). The authors thank Professor Lawrence D. Brown for sharing the call center data. They also thank the editor, the associate editor, and the three referees for valuable comments.

Journal of the American Statistical Association
March 2011, Vol. 106, No. 493, Theory and Methods
DOI: 10.1198/jasa.2011.tm10113

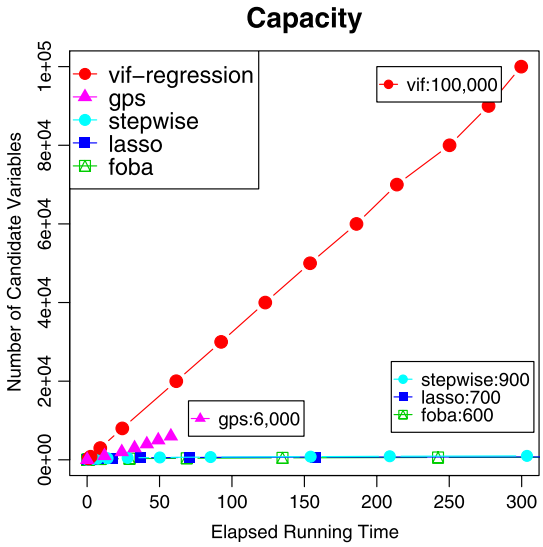

Figure 1. Number of candidate variables examined ("capacity") of five algorithms: VIF regression, stepwise regression, Lasso, FoBa, and GPS, within fixed time (in seconds). The algorithms were asked to search for a model given $n = 1000$ observations and p candidate predictors. VIF regression can run many more variables than any other algorithm; by the 300th second, VIF regression has run 100,000 variables, whereas stepwise regression, Lasso, and FoBa have run 900, 700, and 600, respectively. The implementation of GPS stopped when p is larger than 6000; nevertheless, it is clear that VIF regression can run on much larger data than GPS can. Details of the algorithms and models are given in Section 6. The online version of this figure is in color.

Second, implementations of these algorithms historically have been slow. Our experiments (Section 6) show that Lasso is slow compared with other algorithms; implementation of the Dantzig selector is even slower than the quadratic algorithms (Hastie, Tibshirani, and Friedman 2009), although it can be solved by linear programming. Faster algorithms in this category include coordinate descent (Friedman, Hastie, and Tibshirani 2010) and GPS (Friedman 2008). In Section 6 we show that our algorithm is faster than the fastest of these algorithms, GPS.

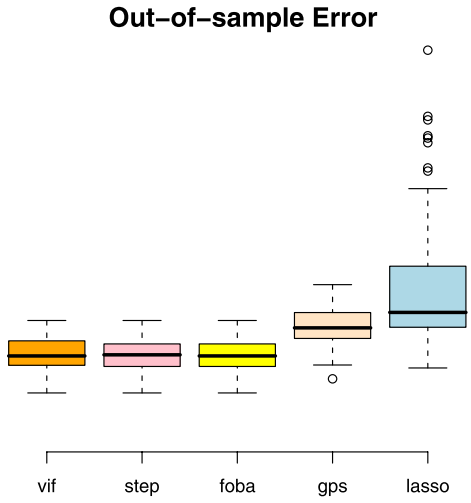

Figure 2. Out-of-sample mean squared errors of the models chosen by the five algorithms. The algorithms were asked to search for a model given $n = 1000$ observations and $p = 500$ independently simulated candidate predictors; mean squared errors of the five chosen models on a test set were computed. We repeated this test 50 times. The figure shows boxplots of these results. VIF regression is as accurate as stepwise regression and FoBa, and much more accurate than GPS and Lasso. The online version of this figure is in color.

More importantly, l_1 algorithms lead to biased estimates (Candes and Tao 2007) and tend to include more spurious variables than l_0 methods, and thus do not perform as well as greedy algorithms in highly sparse systems (Zhang 2009). This bias is due to the fact that these methods minimize a relaxed problem and thus achieve suboptimal solutions to the original problem (Lin et al. 2008). As a result, these optimization algorithms have less accurate predictions; as shown in Figure 10 in Section 6.4, models built by Lasso and GPS are not as accurate as the model fitted using our VIF regression algorithm.

Although efficiently solving nonconvex problems remains highly challenging, progress toward this goal has been reported (Friedman 2008). In the extreme nonconvex case where an l_0 penalty is applied, stepwise regression is still the most accurate approximation algorithm. The VIF regression algorithm that we present in this article is in fact an improved, much faster version of stepwise regression.

1.2 Our VIF Regression Approach

Our VIF algorithm is characterized by the following two components:

- The evaluation step, where we approximate the partial correlation of each candidate variable $\mathbf{x}_i$ with the response variable $\mathbf{y}$ by correcting (using the "variance inflation factor") the marginal correlation using a small presampled set of data. This step can be as fast as $O(n)$ for each variable.
- The search step, in which we test each variable sequentially using an α-investing rule (Foster and Stine 2008). The α-investing rule guarantees no model overfitting and provides highly accurate models.

The evaluation step inherits the spirit of a variation of stepwise regression, *forward stagewise regression*, which evaluates variables only using marginal correlations. The small step-size forward stagewise regression algorithm behaves similarly to l_1 algorithms, such as Lasso and LARS (Efron et al. 2004); thus, like its siblings, it suffers from collinearities among the predictors and will also introduce bias in the estimates. Herein we correct this bias by presampling a small set of data to compute the *variance inflation factor* (VIF) of each variable. The resulting evaluation procedure is fast and does not lose significant accuracy.

This novel VIF procedure can be incorporated with a variety of algorithms, including stepwise regression, LARS, and FoBa. As a demonstration, we incorporate this evaluating procedure with a *streamwise regression* algorithm using an α-investing rule to take full advantage of its speed. Streamwise regression (Zhou et al. 2006), another variation of stepwise regression, considers the case where predictive features are tested sequentially for addition to the model. Because it considers each potential feature only once, it is extremely fast. The resulting VIF regression algorithm is especially useful when feature systems

are dynamically generated and the size of the collection of candidate features is unknown or even infinite. It also can serve as an "online" algorithm to load extremely large-scale data into RAM feature by feature. (Note that our method is available online in *features*, unlike most online regression methods, which are online in observations.)

Our approach is statistics-based in the sense that we add variables only when they are able to pay the price of reducing a statistically sufficient variance in the predictive model. The "price," or the penalty λ_0 in (1), has been well studied in statistics. Classical criteria for the choices include Mallows's C_p, the Akaike information criterion (AIC), the Bayesian information criterion (BIC), the risk inflation criterion (RIC), and many other criteria (Miller 2002). Thus, unlike optimization-based approaches, our algorithm does not require cross-validation.

We compare our VIF algorithm with classic stepwise regression, the Lasso algorithm, and two recently developed algorithms, GPS (Friedman 2008) and FoBa (Zhang 2009). Our experiments produced two main results: (1) The VIF regression algorithm is much faster than any other published algorithms, and (2) the VIF algorithm is comparably accurate to (the slow) stepwise regression and FoBa, but more accurate than (the fast) GPS and Lasso.

The rest of the article is organized as follows. In Section 2 we compare single steps in forward stepwise regression and forward stagewise regression, and show that the coefficient estimate provided by the latter is biased by a factor caused by the multicollinearity and thus needs to be corrected. We propose and present the sped-up streamwise algorithm in Section 3 and note that our algorithm avoids overfitting; it controls the mFDR. In Section 4 we discuss the choice of subsample size, which determines the speed of the algorithm. Section 5 provides guarantees against underfitting, proving that the necessary high signal predictors will not be missed. Finally, in Sections 6 and 7 we experimentally compare VIF against competing methods on several datasets.

2. FORWARD SELECTION AND BIAS CORRECTION

2.1 Forward Feature Selection

Optimally solving (2) with an l_0 penalty requires searching over all 2^p possible subsets, which is NP hard (Natarajan 1995) and thus computationally expensive even when p is small. Computationally tractable selection procedures have been designed to overcome this problem in light of model sparsity and the fact that a majority of the subset models can be ignored. Stepwise regression is such an algorithm.

Stepwise regression sequentially searches for predictors that collectively have strong predictivity. In each step, a multivariate model is statistically analyzed, and a new variable may be added to or an existing variable may be removed from the current model. Common procedures include *forward selection*, *backward elimination*, and a *forward–backward combination*. Forward selection starts from a constant term $\mathbf{1}_n$ and adds one predictor at a time; backward elimination starts from the full set of predictors and removes one predictor in each step. Both procedures have advantages and disadvantages. For data mining applications, however, backward algorithms are unrealistic because of the computational complexity of building models with enormous number of potential explanatory variables. In contrast, forward procedures are much faster, and thus more desirable.

Because multiple regression is needed for *each* candidate predictor in forward stepwise regression, $O(npq^2)$ computation is required for *each* step, where q is the number of variables included in the current model. We assume $p \gg n$. Given the vast set of potential predictors involved, substantial CPU time is often required; thus constructing a more efficient algorithm that can reduce the computational complexity is attractive.

In contrast, in forward *stagewise* regression, only marginal estimates, not partial estimates, are computed in each evaluation step. Thus only $O(np)$ computation is needed, and this procedure is much faster than forward stepwise regression.

We next show that forward stagewise regression leads to a bias that must be corrected to achieve optimal performance. The correction of this bias with be the core of our VIF method.

2.2 Bias Correction

To show that the stagewise evaluation procedure is biased, consider a scheme in which k predictors have already been added to the model, and we are searching for the $k+1$st predictor. Without loss of generality, assume that all of the predictors are *centered* and *normalized*. Because our goal is to find a collectively predictive linear model, we want to test the following alternative hypothetical model:

$$\mathbf{y} = \beta_0 + \beta_1 \mathbf{x}_1 + \cdots + \beta_k \mathbf{x}_k + \beta_{\text{new}} \mathbf{x}_{\text{new}} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathrm{N}(\mathbf{0}, \sigma^2 \mathbf{I}), \quad (3)$$

where $\mathbf{1}_n, \mathbf{x}_1, \ldots, \mathbf{x}_k$ are linearly independent variables. We abuse the notation and remain using σ^2 to denote the variance of the errors. Note that this σ^2 might be different from the more general one presented in Section 1. Denote $\mathbf{X} = [\mathbf{1}_n \ \mathbf{x}_1 \ \cdots \ \mathbf{x}_k]$, $\tilde{\mathbf{X}} = [\mathbf{X} \ \mathbf{x}_{\text{new}}]$, $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_k)'$, and $\tilde{\boldsymbol{\beta}} = (\beta_0, \ldots, \beta_k, \beta_{\text{new}})'$.

Let $\hat{\beta}_{\text{new}}$ be the least squares estimate of β_{new} in model (3). Let $\mathbf{r}$ be the residual of projecting $\mathbf{y}$ on $\{\mathbf{1}_n\} \cup \{\mathbf{x}_i\}_{i=1}^k$. The hypothetical model being considered in stagewise regression is

$$\mathbf{r} = \gamma_{\text{new}} \mathbf{x}_{\text{new}} + \tilde{\boldsymbol{\varepsilon}}, \qquad \tilde{\boldsymbol{\varepsilon}} \sim \mathrm{N}(\mathbf{0}, \tilde{\sigma}^2 \mathbf{I}). \quad (4)$$

We let $\hat{\gamma}_{\text{new}}$ be the least squares estimate of γ_{new} in this model (4) and have the following proposition.

Proposition 1. Under model (3),

$$\hat{\gamma}_{\text{new}} = \rho^2 \hat{\beta}_{\text{new}}, \quad (5)$$

where

$$\begin{aligned} \rho^2 &= \mathbf{x}'_{\text{new}} (\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') \mathbf{x}_{\text{new}} \\ &= \langle \mathbf{x}_{\text{new}}, \mathrm{P}_{\mathbf{X}}^{\perp} \mathbf{x}_{\text{new}} \rangle = \langle \mathrm{P}_{\mathbf{X}}^{\perp} \mathbf{x}_{\text{new}}, \mathrm{P}_{\mathbf{X}}^{\perp} \mathbf{x}_{\text{new}} \rangle \end{aligned} \quad (6)$$

and $\mathrm{P}_{\mathbf{X}}^{\perp}$ is the projection onto the orthogonal complement of the hyperplane spanned by $\{\mathbf{1}_n, \mathbf{x}_1, \ldots, \mathbf{x}_k\}$, in the space spanned by $\{\mathbf{1}_n, \mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{x}_{\text{new}}\}$.

Proof. First, note that

$$\begin{aligned} \tilde{\mathbf{X}}'\tilde{\mathbf{X}} &= \begin{pmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{x}_{\text{new}} \\ \mathbf{x}'_{\text{new}}\mathbf{X} & \mathbf{x}'_{\text{new}}\mathbf{x}_{\text{new}} \end{pmatrix}, \\ (\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1} &= \begin{pmatrix} * & ** \\ -\rho^{-2}\mathbf{x}'_{\text{new}}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} & \rho^{-2} \end{pmatrix}, \end{aligned} \quad (7)$$

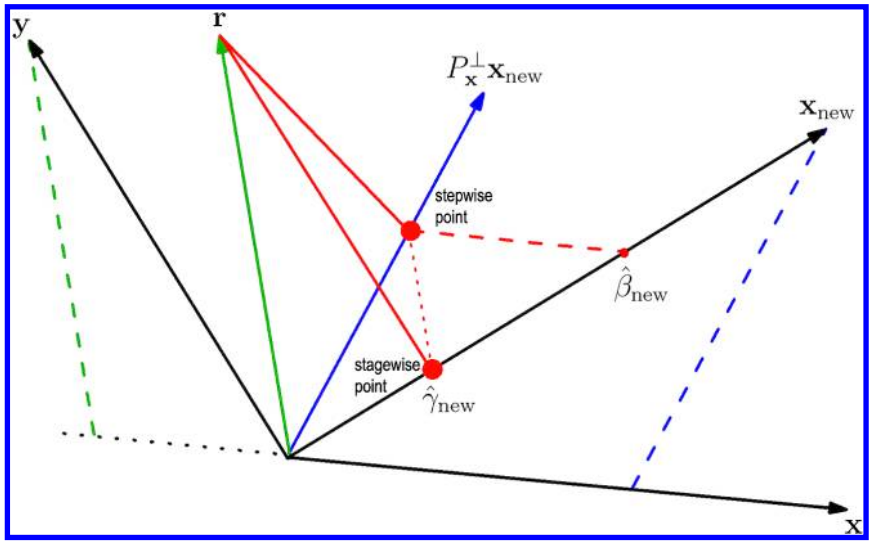

Figure 3. Schematic illustration of Proposition 1. Suppose that $\mathbf{y} = \beta\mathbf{x} + \beta_{\text{new}}\mathbf{x}_{\text{new}} + \boldsymbol{\varepsilon}$. Let $\mathrm{P}_{\mathbf{x}}$ denote the projector on $\mathbf{x}$; then $\mathbf{r} = \mathbf{y} - \mathrm{P}_{\mathbf{x}}\mathbf{y}$ and $P_{\mathbf{x}}^{\perp}\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{new}} - \mathrm{P}_{\mathbf{x}}\mathbf{x}_{\text{new}}$. In stepwise regression, the model fit is the projection of $\mathbf{r}$ on $P_{\mathbf{x}}^{\perp}\mathbf{x}_{\text{new}}$ whereas in stagewise regression, the model fit is the projection of $\mathbf{r}$ on $\mathbf{x}_{\text{new}}$. Note that the red dotted line is perpendicular to $\mathbf{x}_{\text{new}}$, and the red dashed line is perpendicular to $P_{\mathbf{x}}^{\perp}\mathbf{x}_{\text{new}}$, $\hat{\gamma}_{\text{new}}/\hat{\beta}_{\text{new}} = \langle \mathbf{x}_{\text{new}}, \mathrm{P}_{\mathbf{x}}^{\perp}\mathbf{x}_{\text{new}}\rangle^2/\|\mathbf{x}_{\text{new}}\|^2\|\mathrm{P}_{\mathbf{x}}^{\perp}\mathbf{x}_{\text{new}}\|^2 = \langle \mathbf{x}_{\text{new}}, \mathrm{P}_{\mathbf{x}}^{\perp}\mathbf{x}_{\text{new}}\rangle = \rho^2$.

where

$$* = (\mathbf{X}'\mathbf{X})^{-1} + \rho^{-2}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{x}_{\text{new}}\mathbf{x}_{\text{new}}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

and $** = -\rho^{-2}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{x}_{\text{new}}$. Thus,

$$\begin{aligned}\hat{\beta}_{\text{new}} &= (\tilde{\mathbf{X}}'\tilde{\mathbf{X}})_{\text{new}}^{-1}\tilde{\mathbf{X}}'\mathbf{y} \\ &= -\rho^{-2}\mathbf{x}_{\text{new}}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} + \rho^{-2}\mathbf{x}_{\text{new}}'\mathbf{y} \\ &= \rho^{-2}\mathbf{x}_{\text{new}}'\mathbf{r} = \rho^{-2}\hat{\gamma}_{\text{new}}.\end{aligned}$$

A simple case with two variables, shown in Figure 3, illustrates the underlying geometric mechanism of Proposition 1.

Proposition 1 suggests that the stagewise coefficient estimate $\hat{\gamma}_{\text{new}}$ is simply a scaled stepwise coefficient estimate $\hat{\beta}_{\text{new}}$. Thus, if the predictors are all centered, both of the hypothesis tests, $H_0 : \beta_{\text{new}} = 0$ and $H_0 : \gamma_{\text{new}} = 0$, can detect whether or not $\mathbf{x}_{\text{new}}$ contributes to the model. The amount of the contribution that is detected by these two tests is fundamentally different, however.

Under model (3), the expected estimated variance of $\hat{\beta}_{\text{new}}$ is

$$E[\widehat{\text{Var}(\hat{\beta}_{\text{new}})}] = E[(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})_{\text{new}}^{-1}\hat{\sigma}_{\text{step}}^2] = \rho^{-2}\sigma^2$$

by (7), where $\hat{\sigma}_{\text{step}}^2 = (\|\mathrm{r}\|^2 - \rho^{-2}(\mathbf{x}_{\text{new}}'\mathbf{r})^2)/(n-k-2)$ is the mean squared error of this model.

On the other hand, under model assumption (4),

$$E[\widehat{\text{Var}(\hat{\gamma}_{\text{new}})}] = E[\hat{\sigma}_{\text{stage}}^2] = \tilde{\sigma}^2,$$

where $\hat{\sigma}_{\text{stage}}^2 = (\|\mathrm{r}\|^2 - (\mathbf{x}_{\text{new}}'\mathbf{r})^2)/(n-1)$ is the mean squared error of model (4).

Therefore, we have approximately

$$\widehat{\text{Var}(\hat{\gamma}_{\text{new}})} \approx \rho^2\widehat{\text{Var}(\hat{\beta}_{\text{new}})}. \tag{8}$$

It follows that the corresponding t-ratios satisfy

$$t_{\text{new}}^{(\text{stagewise})} \approx |\rho| \cdot t_{\text{new}}^{(\text{stepwise})}. \tag{9}$$

The simulation results shown in Figure 4 demonstrate that these two t-ratios differ by a factor of approximately ρ.

This bias is caused by the misspecified model assumption: under model (3), model (4) is not valid. If $\rho^2 = 1$, $\mathbf{x}_{\text{new}}$ is orthogonal to $\mathbf{X}$, and these two procedures are identical; however, if $\rho^2 < 1$, or $\mathbf{x}_{\text{new}}$ is correlated with $\mathbf{X}$, the errors in model (4) should be correlated. In the latter case, the common model hypothesis testing, which assumes error independence, will not lead to a correct conclusion.

To some extent, forward stepwise regression provides a more powerful procedure in the sense that predictors that can be detected by stagewise regression will be spotted by stepwise regression as well, but not necessarily vice versa. In contrast, the forward stagewise procedures may prefer a spurious predictor

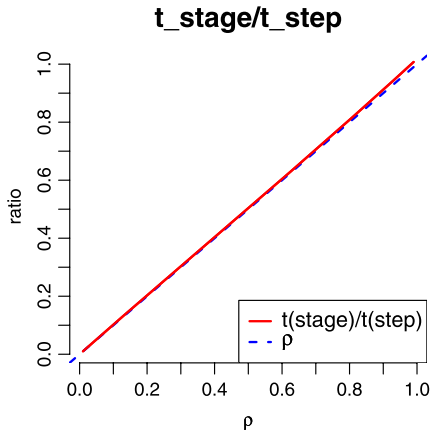

Figure 4. The biased t-ratio. We simulated $\mathbf{y} = \mathbf{x} + \mathbf{x}_{\text{new}} + \mathrm{N}(0, 1)$ with sample size $n = 30$, $\text{Corr}(\mathbf{x}, \mathbf{x}_{\text{new}}) = \sqrt{1-\rho^2}$. For each ρ varying from 0 to 1, we computed both t-statistics of the estimated coefficient of $\mathbf{x}_{\text{new}}$, t_{stage}, and t_{step}, from the two procedures. The ratio $t_{\text{stage}}/t_{\text{step}}$ on ρ is shown. It matches ρ well, as suggested by (9). The online version of this figure is in color.

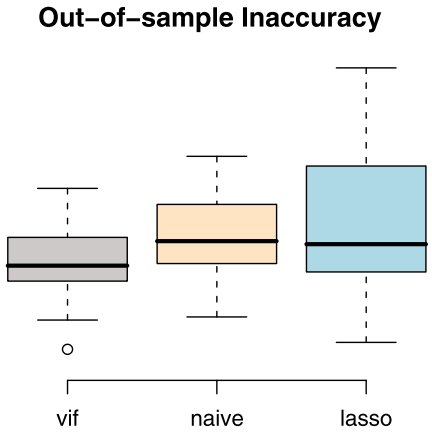

Figure 5. Out-of-sample errors of three algorithms. A naïve algorithm without correction may not be as accurate. The online version of this figure is in color.

that is less correlated with $\mathbf{X}$ to a predictable variable that is highly correlated with $\mathbf{X}$. One of the criticisms of forward selections is that they can never correct the mistakes in earlier steps (Zhang 2009); the inclusion of this spurious variable in the model might lead to more bias. If the data have strong multicollinearity, then the stagewise algorithm will result in a model that is not as predictive.

To illustrate this fact, we simulated $p = 200$ features that are jointly Gaussian and with a covariance matrix of the form (17), with $\theta = 0.9$ and $\tau^2 = 0.1$. The way in which we simulated the response variable $\mathbf{y}$ is similar to the simulations presented in Section 6.3. We compared two algorithms: the VIF regression algorithm that we propose in Section 3 and a *Naïve* algorithm that is exactly the same as the VIF regression algorithm except that it does not include the t-statistic *correction* procedure.

Over 50 replications, we found that on average, VIF regression chose 91% of the true variables, whereas the naïve algorithm chose 47.3% of the true ones. Figure 5 showed the out-of-sample error rate of these two algorithms and Lasso on the same sets of data. It is obvious that the naïve algorithm without a correction procedure does not perform as well as an algorithm based on the corrected statistics.

2.3 The Fast Evaluation Procedure

To speed up the evaluation procedure, we take advantage of the economical computation of forward stagewise regression, but correct the biased t-ratio in each step, thus giving results similar in accuracy to the stepwise regression procedures. Toward this end, we need to estimate the true sample distribution of $\hat{\gamma}_{\text{new}}$ under model (3):

Proposition 2. Under model assumption (3),

$$\hat{\gamma}_{\text{new}} \sim \mathrm{N}(\rho^2 \beta_{\text{new}}, \rho^2 \sigma^2). \tag{10}$$

Proof. Because, by (7), $\hat{\beta}_{\text{new}} \sim \mathrm{N}(\beta_{\text{new}}, \rho^{-2}\sigma^2)$, it follows by Proposition 1.

Now that $\hat{\gamma}_{\text{new}}/(|\rho|\sigma) \sim \mathrm{N}(0, 1)$, with proper estimates of ρ and σ, we can have an honest t-ratio for testing whether or not $\beta_{\text{new}} = 0$:

- $\hat{\sigma}$ can be estimated by the root mean squared error (RMSE), $\hat{\sigma}_{\text{null}}$, under the null model $H_0 : \beta_{\text{new}} = 0$. Unlike $\hat{\sigma}_{\text{step}}$ or $\hat{\sigma}_{\text{stage}}$ (Section 2.2), which are the common estimated standard deviations in regression analysis, using this null estimate $\hat{\sigma}_{\text{null}}$ can prevent overfitting or introducing selection bias, especially in data with heteroscedasticity (Foster and Stine 2004).
- $\hat{\rho}$: ρ can be calculated precisely by proceeding with a multiple regression of $\mathbf{x}_{\text{new}}$ on $\mathcal{C} = \{\mathbf{1}_n, \mathbf{x}_1, \ldots, \mathbf{x}_k\}$, then computing $\rho^2 = 1 - R^2_{\text{new}|1\cdots k}$, the unexplained proportion of variation. But this computation is as expensive as the stepwise procedure, and thus is not desirable. Unfortunately, there is no easy way to estimate ρ because of the dependence issue that we discussed earlier. Most tools, including the bootstrap, break down because of dependency among the errors, which are the only numerical products after stagewise regression is performed. Our solution to this problem is to randomly sample a size m subset of the whole dataset and use this subset to estimate ρ^2 in light of the fact that *each random subset should represent the whole data*. We discuss the choice of m in Section 4.

Our fast evaluation procedure is summarized as follows:

The Fast Evaluation Procedure. At each step of the regression, suppose that a set of predictors, $\mathcal{C} = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, has been chosen in the model. We assume that all of the variables $\mathbf{x}_i$ are centered.

1. Obtain residuals $\mathbf{r} = \mathbf{y} - \mathbf{X}_{\mathcal{C}}(\mathbf{X}'_{\mathcal{C}}\mathbf{X}_{\mathcal{C}})^{-1}\mathbf{X}'_{\mathcal{C}}\mathbf{y}$ and RMSE $\hat{\sigma}_{\text{null}} = \|\mathbf{r}\|/\sqrt{(n - |\mathcal{C}| - 1)}$ from the previous step.
2. Sample a small subset $\mathcal{I} = \{i_1, \ldots, i_m\} \subset \{1, \ldots, n\}$ of observations; let $\mathbf{x}^{\mathcal{I}}$ denote the corresponding subsample of predictors $\mathbf{x}$.
3. Fit $\mathbf{r}$ on $\mathbf{x}_{\text{new}}/\|\mathbf{x}_{\text{new}}\|$ and compute the coefficient estimate $\hat{\gamma}_{\text{new}} = \langle \mathbf{r}, \mathbf{x}_{\text{new}} \rangle / \|\mathbf{x}_{\text{new}}\|$.
4. Fit $\mathbf{x}^{\mathcal{I}}_{\text{new}}$ on $\{\mathbf{x}^{\mathcal{I}}_1, \ldots, \mathbf{x}^{\mathcal{I}}_k\}$ and compute $R^2_{\mathcal{I}} = \mathbf{x}'_{\text{new}}\mathbf{X}^{\mathcal{I}}_{\mathcal{C}} \times ((\mathbf{X}^{\mathcal{I}}_{\mathcal{C}})'\mathbf{X}^{\mathcal{I}}_{\mathcal{C}})^{-1}(\mathbf{X}^{\mathcal{I}}_{\mathcal{C}})'\mathbf{x}_{\text{new}}/\|\mathbf{x}_{\text{new}}\|^2$.
5. Compute and return the approximate t-ratio as $\hat{t}_{\text{new}} = \hat{\gamma}_{\text{new}}/(\hat{\sigma}\sqrt{1 - R^2_{\mathcal{I}}})$.

3. VIF REGRESSION

The fast evaluation procedure can be adapted to speed up a variety of stepwise-like algorithms, but it is most beneficial in massive data settings. Therefore, we incorporate it into a streamwise variable selection algorithm using an α-investing rule.

3.1 α-Investing, Sequential Testing, and mFDR

An α-investing rule is an adaptive, sequential procedure for testing multiple hypotheses (Foster and Stine 2008). The rule works as follows. Suppose that this is a game with a series of tests. A gambler begins his game with initial wealth, w_0; intuitively, this is an allowance for type I error. In the ith test (game), at level α_i, if a rejection is made, then the gambler earns a *pay-out* Δw; otherwise, his current wealth w_i will be reduced by $\alpha_i/(1 - \alpha_i)$. The test level α_i is set to be $w_i/(1 + i - f)$, where f is the time at which the last hypothesis was rejected. Thus, once the gambler successfully rejects a null hypothesis,

he earns more to spend the next few times. Furthermore, the game becomes easier to play in the near future, in the sense that α_i will remain inflated in the short term. The game continues until the player goes bankrupt, that is, $w_i \leq 0$.

The α-investing rule naturally implements a Bonferroni rule, but overcomes its conservativity, controlling instead the mFDR.

The false discovery rate (FDR) aims to control the family-wise error rate (FWER) arising in multiple statistical inferences (Benjamini and Hochberg 1995). In multiple hypothesis testing, the successful rejection of a null hypothesis is called a *discovery*. The classic definition of FDR is the expected proportion of false discoveries among all discoveries throughout the whole process,

$$\text{FDR} = E\left(\frac{V}{R}\middle| R > 0\right)P(R > 0), \tag{11}$$

where V is the number of false positives and R is the number of total discoveries. A few variants of FDR have been introduced in the past decade, including the mFDR, which is defined as $E(V)/E(R)$ or $E(V)/(E(R) + 1)$; the positive false discovery rate (pFDR) (Storey 2002), which drops the term $P(R > 0)$ in (11); and the local false discovery rate (fdr) (Efron et al. 2001), which is determined by the size of the test statistic z.

An α-investing procedure controls mFDR in a sequential setting (Foster and Stine 2008).

Proposition 3. An α-investing rule with initial alpha-wealth $w_0 \leq \alpha\eta$ and pay-out $\Delta w \leq \alpha$ controls $\text{mFDR}_\eta = E(V)/(E(R) + \eta)$ at level α.

See Foster and Stine (2008) for the technical details of this theorem.

3.2 Steamwise Variable Selection and VIF Regression

Using an α-investing rule allows us to test an infinite stream of hypotheses while controlling mFDR. In the context of variable selection, this implies that we may order the variables in a sequence (possibly dynamically) and include them into the model in a streamwise manner *without overfitting*.

Overfitting is a common problem in regression analysis. The model R^2 will increase when a new variable is added, regardless of whether or not it is spurious. This in-sample overfitting may result in terrible predictions when the model is used out of sample. Thus the goal of all variable selection problems is to find a *parsimonious* model that has a satisfactory R^2 or model fit, to avoid overfitting. These problems will typically impose a penalty on the number of variables in the model, namely the l_0 norm of the coefficient parameters, as we introduced in Section 1. Forward selection approaches the solutions to these problems by properly thresholding the t-ratios of upcoming variables to control the number of selected variables.

The ability to test the variables in a streamwise way has many advantages. First, the one-pass algorithm can save a great amount of computation if the data are massive. In most search algorithms, adding *each* new variable necessitates going through the whole space of candidate variables; the computation is expensive if the data size, $n \times p$, is huge. We alleviate this burden by reducing the loops to only one round. Second, this allows one to handle dynamic variable sets. These include the cases where p is extremely large or unknown, resulting in a problem in applying static variable selection criteria. This also allows one to first test the lower-order interactions and then decide which higher-order interactions need to be tested.

Algorithm 1 VIF regression. The boosted Streamwise Regression using α-investing

Input: data $\mathbf{y}, \mathbf{x}_1, \mathbf{x}_2, \ldots$ (centered);
Set: initial wealth $w_0 = 0.50$ and pay-out $\Delta w = 0.05$, and subsample size m;
Initialize $\mathcal{C} = \{0\}$; $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$; $\hat{\sigma} = \text{sd}(\mathbf{y})$; $i = 1$; $w_1 = w_0$; $f = 0$.
Sample $\mathcal{I} = \{i_1, \ldots, i_m\} \subset \{1, \ldots, n\}$.
repeat
  set threshold $\alpha_i = w_i/(1 + i - f)$
  attain $\hat{t}_i$ from the Fast Evaluation Procedure // compute corrected t-statistic
  if $2\Phi(|t_i|) > 1 - \alpha_i$ // compare p-value to threshold **then**
    $\mathcal{C} = \mathcal{C} \cup \{i\}$ // add feature to model
    update $\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}_\mathcal{C}$, $\hat{\sigma} = \text{RMSE}_\mathcal{C}$
    $w_{i+1} = w_i + \Delta w$
    $f = i$
  else
    $w_{i+1} = w_i - \alpha_i/(1 - \alpha_i)$
  end if
  $i = i + 1$
until maximum CPU time or Memory is reached
*Φ is the CDF of the normal distribution.

Given the α-investing rule for sequential variable selection, we may proceed with our algorithm in a streamwise way with a guarantee of no overfitting. We state our VIF regression procedures in Algorithm 1. We call it "VIF" because the correction factor ρ in the key speed-up part is the *variance inflation factor* of the new variable with respect to the included variables.

One might worry that going through the candidate predictors only once may miss signals. In the worst case, it may in fact miss useful predictors; however, this will not occur in cases where the variables are orthogonal, as in, for example, orthogonally designed experiments and signal processing (using a Fourier transform or wavelet transform). This also applies to distributionally orthogonal variables, as in, for example, independent Gaussian entries in image processing. If predictors are highly correlated, then each of these variables may contribute to the model, because we are looking for a collectively linear model. As we prove later, using an α-investing rule in this case guarantees that the final model will have certain predictability. Our experiments (Section 6) show that the test accuracy of the models chosen by the VIF algorithm is highly competitive with that of models chosen by the most accurate algorithms for linear models. Furthermore, if we have previous knowledge of the predictors (e.g., for PCA variables), then we can assign a higher priority to important variables so that they can be entered into the model more easily.

4. ACCURACY AND COMPUTATIONAL COMPLEXITY

Obviously, a large m (i.e., many observations used to test for inclusion of a feature) can guarantee an accurate approximation in our algorithm (Algorithm 1), but a small m will provide faster

computation. How large should m be to attain a reasonably accurate result? Ideally, we want to pick $m \ll n$ and small α and ϵ, such that

$$P\left(\left|\frac{|\hat{\rho}| - |\rho|}{|\rho|}\right| \le \epsilon \middle| \rho\right) \ge 1 - \alpha,$$

where ρ is defined as in (6), the correlation between $\mathbf{x}_{\text{new}}$ and the perpendicular space of the space spanned by included variables, and $\hat{\rho}$ is the sample correlation between $\mathbf{x}_{\text{new}}^{\mathcal{I}}$ and $\text{span}\{\mathbf{1}_m, \mathbf{x}_1^{\mathcal{I}}, \ldots, \mathbf{x}_k^{\mathcal{I}}\}^{\perp}$. This implies that with high probability, the bias in the correlation due to the subsampling is not large compared with the true correlation. Then, roughly with probability at least $1 - \alpha$, the approximate t-ratio is

$$|\hat{t}| = \frac{|\hat{\gamma}_{\text{new}}|}{\hat{\sigma}\,|\hat{\rho}|} = \frac{|\hat{\gamma}_{\text{new}}|}{\hat{\sigma}\,|\rho|(1 + (|\hat{\rho}| - |\rho|)/|\rho|)}$$
$$\approx \frac{|\hat{\gamma}_{\text{new}}|}{\hat{\sigma}\,|\rho|}\left(1 - \frac{|\hat{\rho}| - |\rho|}{|\rho|}\right).$$

Consequently, with probability at least $1 - \alpha$,

$$(1 - \epsilon)|t_{\text{true}}| \lessapprox |\hat{t}| \lessapprox (1 + \epsilon)|t_{\text{true}}|. \tag{12}$$

Recall that $\rho^2 = 1 - R^2_{\text{new}|1\cdots k}$. Let $\mathbf{z} = \text{P}_{\mathbf{X}}^{\perp}\mathbf{x}_{\text{new}}$, where the operator $\text{P}_{\mathbf{X}}^{\perp}$ is defined as in Proposition 1. Then ρ is the sample correlation of $\mathbf{x}_{\text{new}}$ and $\mathbf{z}$. Also assume that $(\mathbf{x}_{\text{new}}, \mathbf{z})$ are random iid samples from a bivariate normal population with correlation ρ_0. Then, approximately,

$$\frac{1}{2}\log\left(\frac{1+\rho}{1-\rho}\right) \overset{\text{approx}}{\sim} \text{N}\left(\frac{1}{2}\log\left(\frac{1+\rho_0}{1-\rho_0}\right), \frac{1}{n-3}\right).$$

Thus, conditional on the observations (and due to the fact that we sample without replacement), we have approximately

$$\frac{1}{2}\log\left(\frac{1+\hat{\rho}}{1-\hat{\rho}}\right)\Big|\rho \overset{\text{approx}}{\sim} \text{N}\left(\frac{1}{2}\log\left(\frac{1+\rho}{1-\rho}\right), \frac{1}{m-3}\right). \tag{13}$$

Because we focus on datasets with huge n's and in high-dimensional spaces, it is unlikely that two random vectors would be highly correlated. In fact, we can show that a d-dimensional space can tolerate up to $O(d^2)$ random vectors with angles exceeding $\pi/4$. In light of this fact and the approximate sample distribution (13), a crude calculation by assuming $|\rho| > \sqrt{2}/2$ shows that $m \ge 200$ can guarantee an $\epsilon \le 0.1$ and an $\alpha \le 0.05$ in (12).

As a numerical example, we examined the Boston housing dataset, which contains 506 census tracts in Boston from the 1970 census. These data and their description can be downloaded from the UCI Repository of Machine Learning Databases at *http://archive.ics.uci.edu/ml/*. We took `MEDV`, the median value of owner-occupied homes, as our response variable. We then sequentially added the other 13 variables in a multiple linear regression model as explanatory variables. In each step, we computed the "true" t-ratio t_{true} of the incoming variable by replacing the new RMSE with the old one (see Section 2.3). In addition, we repeated subsampling with size $m = 200$ and our fast evaluation procedure 100 times, resulting in 100 fast t-ratios $|\tilde{t}|$. We then collected the ratios $|\tilde{t}|/|t_{\text{true}}|$.

Figure 6 shows a comparative boxplot summarizing these experimental results on the 13 explanatory variables of the Boston housing data. As shown in the boxplot, taking $\epsilon = 0.1$, most of the ratios lie within the interval $[1 - \epsilon, 1 + \epsilon]$. To see how sensitive these bounds are to the actual correlation, we computed $|\rho|$ based on Proposition 1; these $|\rho|$'s are annotated under the corresponding variables in Figure 6 and are also listed in Table 1. Several variables have $|\rho|$ less than $\sqrt{2}/2$. For these variables, despite high variances, the ratios of absolute t-ratios are well bounded by $1 \pm 15\%$. This experiment validates our earlier claim that with a subsample size of $m = 200$, our fast evaluation mechanism can provide a tight bound on the accuracy in terms of the t-ratio approximation.

Because VIF regression does a single pass over the predictors, it has a total computational complexity of $O(pmq^2)$, where m is the subsample size and q is the number of variables in the final model. Assuming sparsity in the model found, q can be much smaller than n; thus, as long as $m = O(n/q^2)$, which

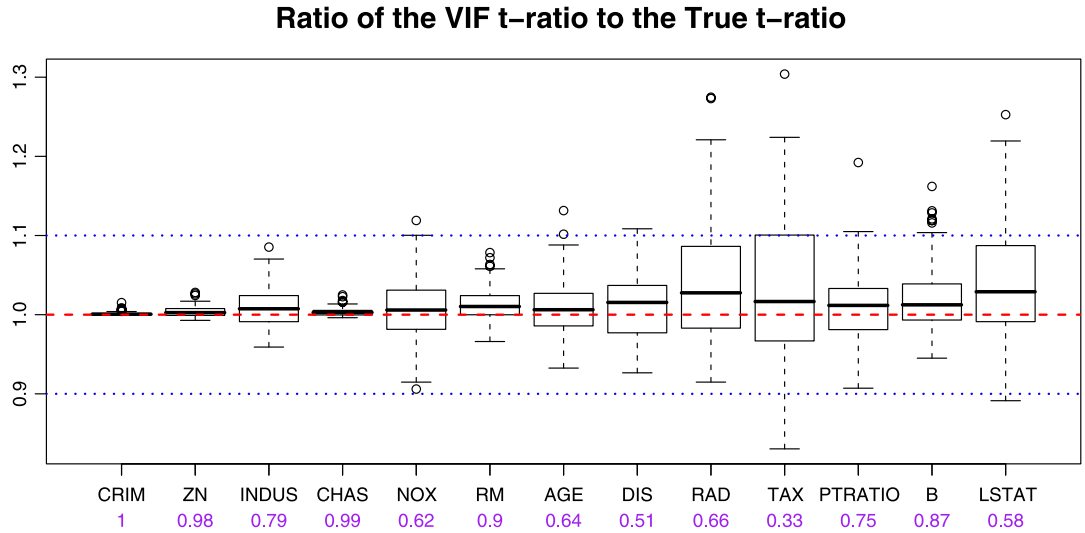

Figure 6. Simulation of $|\hat{t}|$ for the Boston housing data. We added these variables into our multiple linear regression model sequentially. For each variable, the approximate t-ratio $|\hat{t}| = |\hat{\gamma}_{\text{new}}|/\hat{\sigma}\,|\hat{\rho}|$ was computed based on a subsample of size $m = 200$. These boxplots result from a simulation of 100 subsample sets. Annotated below the variables are the true $|\rho|$'s. The online version of this figure is in color.

Table 1. True $|\rho|$'s in the Boston housing data. We added these variables into our multiple linear regression model sequentially. The $|\rho|$ values when the corresponding variable is added in the model are displayed. These $|\rho|$'s are computed using (6)

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE
1.00	0.98	0.79	0.99	0.62	0.90	0.64
DIS	RAD	TAX	PTRATIO	B	LSTAT	
0.51	0.66	0.33	0.75	0.87	0.58	

can be easily achieved based on our earlier discussion, the total computational complexity is $O(pn)$.

5. STABILITY

Proposition 3 guarantees that our algorithm will not overfit the data. In this section we develop a theoretical framework and show that our algorithm will not miss important signals.

A locally important variable gets added into the model if its reduction to the sum of squared errors exceeds the penalty λ that it brings to the penalized likelihood. However, if this importance can be washed out or masked by other variables, then, for prediction purposes, there is no difference between this variable and its surrogates, and thus neither of them can be claimed "true." This situation is common in our application, because we consider predictors that are correlated or even highly correlated by including high-order interactions. Predictive accuracy will be lost only when those globally important variables, which stand out in any scenarios, are missed. Toward this end, we propose the following theorem, which guarantees that none of these important variables will be missed.

Let $\mathcal{M}$ be the subset of nonconstant variables that are currently chosen. We define

$$\mathcal{S}_{\lambda,\eta}(\mathcal{M}) = \left\{ \mathbf{x} : \frac{\mathrm{SSE}_{\mathcal{M}} - \mathrm{SSE}_{\mathcal{M}\cup\mathbf{x}}}{\mathrm{SSE}_{\mathcal{M}}/(n - |\mathcal{M}| - 1)} > (1+\eta)\lambda \right\} \quad (14)$$

as the collection of variables that are λ-*important* with respect to model $\mathcal{M}$ and

$$\mathcal{S}_{\lambda,\eta} = \bigcap_{\mathcal{M}} \mathcal{S}_{\lambda,\eta}(\mathcal{M}) \quad (15)$$

as the collection of λ-*important* variables. Note that both of these are random sets; that is, they depend on the observed data. Let $\hat{\mathcal{C}}_{\mathrm{step}}$, $\hat{\mathcal{C}}_{\mathrm{stream}}$, and $\hat{\mathcal{C}}_{\mathrm{VIF}}$ be the models chosen by stepwise regression, streamwise regression with an α-investing rule, and VIF regression, respectively. An investing rule is called η-patient if it spends at a sufficiently slow rate such that it has enough saved to spend at least $i^{-(1+\eta)}$ on the ith variable. For example, the investing rules in Zhou et al. (2006) and Foster and Stine (2008) can be chosen to be η-patient. We have the following theorem.

Theorem 1. When the algorithms stop,

(1) $\mathcal{S}_{\lambda,0} \subset \hat{\mathcal{C}}_{\mathrm{step}}$.
(2) If the number of candidate predictors $p > 7$ and an η-patient investing rule is used, then $\mathcal{S}_{2\log p,\eta} \subset \hat{\mathcal{C}}_{\mathrm{stream}}$.
(3) Suppose that the $\mathbf{x}$'s are multivariate Gaussian. If we use an $\eta(1-\eta)/2$-patient investing rule and our sampling size m is large enough, then, for any $\mathbf{x} \in \mathcal{S}_{2\log p,\eta}$, we have $P(\mathbf{x} \in \hat{\mathcal{C}}_{\mathrm{VIF}}) > 1 - O(1/m)$.

In other words, any $2\log p$-*important* variable likely will be included by the VIF algorithm.

Proof of Theorem 1. (1) $\forall \mathbf{x} \in \mathcal{S}_{\lambda,\eta}$, if $\mathbf{x} \notin \hat{\mathcal{C}}_{\mathrm{step}}$, then $\mathrm{SSE}_{\hat{\mathcal{C}}_{\mathrm{step}}} + |\hat{\mathcal{C}}_{\mathrm{step}}| \cdot \lambda\hat{\sigma}^2_{\hat{\mathcal{C}}_{\mathrm{step}}} < \mathrm{SSE}_{\hat{\mathcal{C}}_{\mathrm{step}}\cup\mathbf{x}} + (|\hat{\mathcal{C}}_{\mathrm{step}}| + 1) \cdot \lambda\hat{\sigma}^2_{\hat{\mathcal{C}}_{\mathrm{step}}}$, and $\mathrm{SSE}_{\hat{\mathcal{C}}_{\mathrm{step}}} - \mathrm{SSE}_{\hat{\mathcal{C}}_{\mathrm{step}}\cup\mathbf{x}} < \lambda\hat{\sigma}^2_{\hat{\mathcal{C}}_{\mathrm{step}}} = \lambda\mathrm{SSE}_{\hat{\mathcal{C}}_{\mathrm{step}}}/(n - |\hat{\mathcal{C}}_{\mathrm{step}}| - 1)$, which contradicts the definition of $\mathcal{S}_{\lambda,\eta}$.

(2) Suppose that the current model is $\mathcal{M}_0$. If the next predictor $\mathbf{x}_i \in \mathcal{S}_{2\log p,\eta}$, then it has a t-statistic, t_i, that meets

$$\begin{aligned} P(|Z| > |t_i|) &< P(|Z| > \sqrt{(1+\eta)2\log p}) \\ &< \frac{2\exp\{-(1+\eta)2\log p/2\}}{\sqrt{(1+\eta)2\log p}} < \frac{1}{p^{(1+\eta)}} \end{aligned}$$

as long as $p > 7$. Thus $\mathbf{x}$ will be chosen by any η-patient investing rule.

(3) We follow the notation given in Section 4. Suppose that the current model is $\mathcal{M}_0$. Let $\rho = \sqrt{1 - R^2_{\mathbf{x}_i|\mathcal{M}_0}} > 0$ and $\hat{\rho}$ be its VIF surrogate. If the next candidate predictor $\mathbf{x}_i \in \mathcal{S}_{2\log p,\eta}$ has a VIF-corrected t-statistic $\hat{t}_i$ and true t-statistic t_i, we then have

$$\begin{aligned} &P\left(|\hat{t}_i| > \sqrt{\left(1 + \frac{\eta}{2} - \frac{\eta^2}{2}\right)2\log p}\,\middle|\,\mathbf{X}, \mathbf{y}, \mathcal{M}_0\right) \\ &> P\left(|\hat{t}_i| > |t_i|\sqrt{1 - \frac{\eta}{2}}\,\middle|\,\mathbf{X}, \mathbf{y}, \mathcal{M}_0\right) \\ &= P\left(|\hat{\rho}| < \frac{|\rho|}{\sqrt{1 - \eta/2}}\,\middle|\,\rho\right) \\ &= P\left(\hat{\rho}^2 < \frac{\rho^2}{1 - \eta/2}\,\middle|\,\rho\right) \\ &> P\left(\hat{\rho}^2 < \rho^2\left(1 + \frac{\eta}{2}\right)\,\middle|\,\rho\right) \\ &> 1 - \tilde{m}^{-1/2}\frac{8(1-\rho^2) + \eta}{2\eta\rho}\phi(\kappa) + \tilde{m}^{-1/2}\frac{3\rho^2 - 1}{2\rho}\kappa^2\phi(\kappa) \\ &\quad + \tilde{m}^{-1}\left(\frac{1}{2\rho^2} - 2 + \frac{13}{4}\rho^2\right)\kappa^3\phi(\kappa) \\ &\quad - \tilde{m}^{-1}\frac{(3\rho^2 - 1)^2}{8\rho^2}\kappa^5\phi(\kappa) + O\left(\tilde{m}^{-3/2}\right) \\ &> 1 - O(m^{-1}), \end{aligned} \quad (16)$$

where $\tilde{m} = m - 3/2 + \rho^2/4$, $\kappa = \tilde{m}^{1/2}\eta\rho/4(1-\rho^2)$, $\phi(\cdot)$ is the density function of standard normal distribution, and the expansion in the sixth line follows Konishi (1978), with $m > 16(1-\rho^2)/\rho^2\eta^2 + 2$. Note that $\kappa^3\phi(\kappa)$ is bounded and the first two nonconstant terms are as small as order m^{-1} with sufficiently large m; the third term is always positive, which covers the last two terms. The final bound follows from these.

Several recent articles have addressed the selection consistency of forward selection. Wang (2009) used stepwise regression to screen variables and then performed the common l_1 methods on the screened variables. The author showed that the screening path would include the true subset asymptotically, and thus the consistency of l_1 methods might pertain.

Cai and Wang (2010) used orthogonal matching pursuit, which is essentially a stagewise regression algorithm. They showed that with certain stopping rules, the important variables (with large true β) can be fully recovered with high probability. However, both articles assume near-orthogonality and use parameters to constrain multicollinearity, with bounded eigenvalues in the former and mutual incoherence in the latter. Zhang (2009) made similar assumptions. In our statistical applications, however, multicollinearity is common because we consider interaction terms, and thus such consistency results are of limited utility. Also, as long as multicollinearity exists, there is no proper definition for "true variables," because the significance of one variable might be washed out by other variables. Thus, the best that can be achieved are theorems such as the foregoing guaranteeing that high signal predictors will not be missed if no other predictors are highly correlated with them. If multiple predictors are high signal but correlated, then we will find at least one of them.

6. NUMERICAL EXPERIMENTS

To test the performance of VIF regression, we compare it with the following four algorithms:

- Classic *Stepwise regression*. For the penalty criterion, we use either BIC or RIC, depending on the size of the data.
- *Lasso*, the classic l_1 regularized variable selection method (Tibshirani 1996). Lasso can be realized by the least angle regression (LARS) algorithm (Efron et al. 2004), scaling in quadratic time in the size, n of the data set.
- *FoBa*, an adaptive forward–backward greedy algorithm focusing on linear models (Zhang 2009). FoBa does a forward–backward search; in each step, it adds the most correlated predictor and/or removes the least correlated predictor. This search method is very similar to stagewise regression, except that it behaves adaptively in backward steps. Zhang (2009) provided a theoretical bound on the parameter estimation error.
- *GPS*, the generalized path-seeking algorithm (Friedman 2008). GPS is a fast algorithm that finds ℓ_ϵ regularized models via coordinate descent. For $p \ll n$, its computation can be as fast as linear in n (Friedman 2008). GPS can compute models for a wide variety of penalties, and selects the penalty via cross-validation.

In the following sections, we examine different aspects of these algorithms, including speed and performance, on both synthetic and real datasets. All of the implementations were done in R, a widely used statistical software package (available at *http://www.r-project.org/*). We emphasize that, unlike our VIF algorithm and stepwise regression, whose penalties are chosen statistically, the other three algorithms are cast as optimization problems and thus require cross-validation to determine either the penalty function (GPS) or the sparsity (Lasso and FoBa). Because sparsity is generally unknown, to fairly compare these algorithms, we did not specify the sparsity even for synthetic data. Instead, we used five-fold cross-validation for Lasso and GPS and two-fold cross-validation for FoBa. Note that this adds only a constant factor to the computational complexity of these algorithms.

6.1 Design of the Simulations

In each simulation study, we simulated p features, $\mathbf{x}_1, \ldots, \mathbf{x}_p$. We mainly considered three cases of collinearity: (1) the $\mathbf{x}$'s are independent random vectors with each X_{ij} (the jth element of $\mathbf{x}_i$) simulated from N(0, 0.1), that is, the $\mathbf{x}$'s are jointly Gaussian with covariance matrix $\Sigma_1 = \tau^2 \mathbf{I}_p$, where $\tau^2 \equiv 0.1$; (2) the $\mathbf{x}$'s are jointly Gaussian with covariance matrix

$$\Sigma_2 = \tau^2 \begin{pmatrix} 1 & \theta & \cdots & \theta^{p-1} \\ \theta & 1 & \cdots & \theta^{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \theta^{p-1} & \theta^{p-2} & \cdots & 1 \end{pmatrix} \tag{17}$$

with $\tau^2 \equiv 0.1$; and (3) the $\mathbf{x}$'s are jointly Gaussian with covariance matrix

$$\Sigma_3 = \tau^2 \begin{pmatrix} 1 & \theta & \cdots & \theta \\ \theta & 1 & \cdots & \theta \\ \vdots & \vdots & \ddots & \vdots \\ \theta & \theta & \cdots & 1 \end{pmatrix} \tag{18}$$

with $\tau^2 \equiv 0.1$. We randomly picked $q = 6$ variables from these p variables. The response variable $\mathbf{y}$ was generated as a linear combination of these q variables plus a random normal noise. The q predictors have equal weights, $\beta = 1$, in all sections except Section 6.5, where the weights are set to be $\{6, 5, 4, 3, 2, 1\}$. The random normal noise in most sections has mean 0 and variance 1 without further explanation; its variances varies from 0.4 to 4 in Section 6.5 to investigate different signal-to-noise ratios.

In all simulations, we simulated $2n$ independent samples, then used n of them for variable selection and another n for out-of-sample performance testing. The out-of-sample performance was evaluated using the mean sum of squared errors, $\sum_{i=n+1}^{2n} (y_i - \mathbf{x}_i \hat{\boldsymbol{\beta}})^2 / n$, where $\hat{\boldsymbol{\beta}}$ is the output coefficient determined by the five algorithms based on the training set, namely the first n samples. The sample size n is fixed at 1000 without further clarification. Because the true predictors were known, we also compared the true discovery rate and FDR in Section 6.3.

6.2 Comparison of Computation Speed

We simulated the independent case to measure the speed of these five algorithms. The response variable $\mathbf{y}$ was generated by summing six of these features with equal weights plus a random noise N(0, 1). Considering the speed of these five algorithms, the number of features p varies from 10 to 1000 for all five algorithms, and from 1000 to 10,000 for VIF regression and GPS.

As shown in Figure 7, VIF regression and GPS perform almost linearly and are much faster than the other three algorithms. Given the fact that it does a marginal search, the FoBa algorithm is surprisingly slow; thus we did not perform cross-validation for this speed benchmarking.

To further compare VIF and GPS, Figure 8 shows two close-up plots of the running time of these two algorithms. Both plots appear to be linear in p, the number of candidate predictors. Although GPS leads when p is small, VIF regression has a smaller slope and is much faster when p is large.

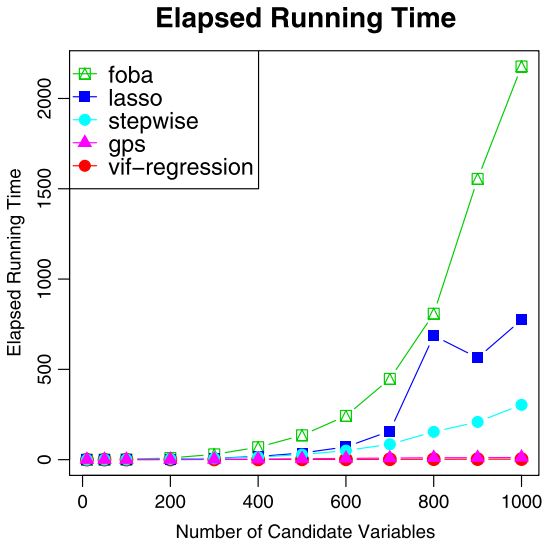

Figure 7. Running time (in seconds) of the five algorithms VIF regression, stepwise regression, Lasso, FoBa, and GPS. The algorithms were asked to search for a model given $n = 1000$ observations and p candidate predictors; p varies from 10 to 1000. The online version of this figure is in color.

Table 2. Summary of the average numbers of true discoveries, false discoveries, and estimated mFDR using the five algorithms in the experiment with independent Gaussian features. The training set contained 1000 observations and p features, six of which were used to create the response variables. This simulation was repeated 50 times

		Methods				
Cases		VIF	Stepwise	FoBa	GPS	Lasso
$p = 100$	True	6.0	6.0	6.0	6.0	5.86
	False	0.82	0.02	0.04	0.18	38.82
	mFDR	0.049	0.001	0.002	0.011	0.710
$p = 200$	True	6.0	6.0	6.0	6.0	5.38
	False	0.56	0.04	0.02	0.08	70.02
	mFDR	0.034	0.002	0.001	0.005	0.820
$p = 300$	True	6.0	6.0	6.0	6.0	5.66
	False	0.60	0.06	0.02	0.04	75.44
	mFDR	0.036	0.004	0.000	0.002	0.828
$p = 400$	True	6.0	6.0	6.0	6.0	5.50
	False	0.56	0.10	0.00	0.02	93.78
	mFDR	0.034	0.006	0.000	0.001	0.858
$p = 500$	True	6.0	6.0	6.0	6.0	5.48
	False	0.58	0.04	0.00	0.04	117.78
	mFDR	0.035	0.002	0.000	0.002	0.884

6.3 mFDR Control

To test whether or not these algorithms successfully control mFDR, we studied the performance of the models chosen by the five algorithms based on the training set. We took the simulation scheme in Section 6.1 and repeated the same simulation 50 times. We then computed the average numbers of false discoveries and true discoveries of features, denoted by $\widehat{E(V)}$ and $\widehat{E(S)}$, respectively. Taking an initial wealth of $w_0 = 0.5$ and a payout of $\Delta w = 0.05$ in our VIF algorithm, with an $\eta = 10$ in Proposition 3, the estimated mFDR is given by

$$\widehat{\text{mFDR}}_\eta = \frac{\widehat{E(V)}}{\widehat{E(V)} + \widehat{E(S)} + \eta}. \tag{19}$$

Table 2 summarizes $\widehat{E(S)}$, the average number of true discoveries; $\widehat{E(V)}$, the average number of false discoveries; and $\widehat{\text{mFDR}}_\eta$, the estimated mFDR, in the first simulation with independent Gaussian features. As can be seen, all algorithms but Lasso successfully spotted the six true variables and controlled mFDR well. This is not surprising, because these algorithms aim to solve nonconvex problems (Section 1). Lasso solves a relaxed convex problem and thus tends to include many spurious variables and then shrinks the coefficients to reduce the prediction risk.

Table 3 provides a similar summary for the case where the features were generated using a multivariate Gaussian distribution with the covariance matrix given in (17). Lasso again was not able to control mFDR. Both stepwise regression and FoBa controlled mFDR at low levels in all cases. GPS and VIF regression also did well except for the case with very high multicollinearity. However, as we mentioned earlier, in the case with high multicollinearity, each of the collinear predictors could make a contribution to model accuracy, because we are building a nested model. Thus it is difficult to claim that the "false discoveries" are indeed false in building a multiple linear model. In any case, because our main purpose in using an α-investing control rule is to avoid model overfitting, we examine their out-of-sample performance in the next section.

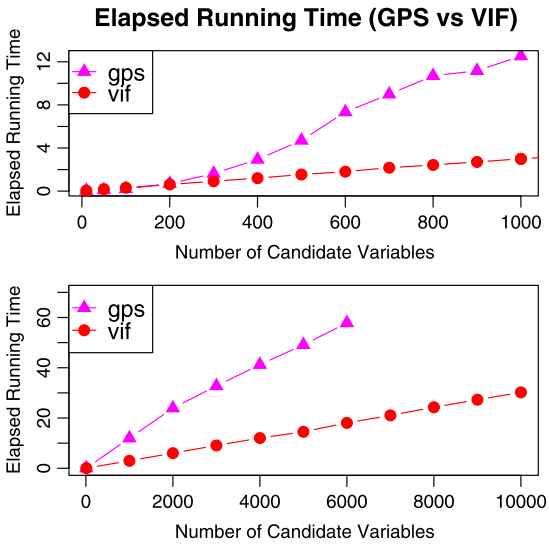

Figure 8. Running time (in seconds) of VIF regression and GPS algorithm. The algorithms were asked to search for a model given $n = 1000$ observations and p candidate predictors; p varies from 10 to 10,000. The online version of this figure is in color.

6.4 Out-of-Sample Performance

We used the aforementioned $n = 1000$ held-out observations to test the models chosen by the five algorithms. The case with independently generated features is illustrated in Figure 9,

Table 3. Summary of the average numbers of true discoveries, false discoveries, and estimated mFDR using the five algorithms in the experiment with jointly Gaussian features. The training set contained 1000 observations and 200 features, six of which were used to create the response variables. The θ in (17) were taken to be 0.1, 0.3, 0.5, 0.7, and 0.9. This simulation was repeated 50 times

		Methods				
Cases		VIF	Stepwise	FoBa	GPS	Lasso
$\theta = 0.1$	True	6.00	6.00	6.00	6.00	5.64
	False	0.56	0.02	0.02	0.26	72.94
	mFDR	0.034	0.001	0.001	0.016	0.823
$\theta = 0.3$	True	6.00	6.00	6.00	6.00	5.54
	False	2.04	0.02	0.02	0.12	68.40
	mFDR	0.113	0.001	0.001	0.007	0.815
$\theta = 0.5$	True	6.00	6.00	6.00	5.90	5.86
	False	6.30	0.04	0.10	0.20	74.12
	mFDR	0.282	0.002	0.006	0.012	0.824
$\theta = 0.7$	True	6.00	6.00	6.00	6.00	5.84
	False	13.20	0.04	0.16	0.60	64.58
	mFDR	0.452	0.002	0.010	0.036	0.803
$\theta = 0.9$	True	5.46	5.66	5.46	5.84	5.90
	False	32.30	0.33	0.64	2.44	76.22
	mFDR	0.676	0.019	0.038	0.133	0.827

which shows a comparative boxplot for the out-of-sample mean squared errors of the five chosen models in 50 runs. As can be seen, the models chosen by VIF regression perform as well as the two best algorithms (stepwise regression and FoBa) and better than GPS and Lasso. Figure 10 provides a similar scenario for jointly Gaussian features, except for the case with extremely high correlation. Here VIF regression has slightly higher mean squared errors but is still better than GPS and Lasso. The latter boxplot clarifies our claim that although VIP regression discovered more "false discoveries," these features were not truly false. In fact, they helped build a multiple model that did not overfit, as shown in Figure 10. In this sense, VIF regression does control mFDR. Given that VIF regression is significantly faster than other algorithms, these results are very satisfactory.

6.5 The Effect of Signal-to-Noise Ratio

To show how the signal-to-noise ratio might affect our algorithm, we took the simulation scheme with Σ_2 and $\theta = 0.5$ or 0.9. The number of features p was fixed as 200. $\mathbf{y}$ was a linear combination of $q = 6$ randomly chosen variables with weights of 1–6, plus an independent random noise, $N(0, \nu^2)$, where ν varies between 0.4 and 4. We used $w_0 = 0.5$ and $\Delta w = 0.05$ for the VIF algorithm.

We computed the out-of-sample mean squared errors on the $n = 1000$ held-out samples. To better illustrate the performance of the five algorithms, we report the ratio of the out-of-sample mean squared errors of other algorithms to that of VIF regression, that is, $\sum_{i=n+1}^{2n}(y_i - \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{other}})^2 / \sum_{i=n+1}^{2n}(y_i - \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{vif}})^2$. A ratio less than (greater than) 1 implies better (worse) performance of the algorithm compared with that of the VIF regression.

In general, VIF regression was slightly worse than stepwise regression and FoBa, but was much better than GPS and Lasso. When the multicollinearity of the variables was weak (with $\theta = 0.5$), as shown in Figure 11, the VIF regression performed almost as well as stepwise regression and FoBa (with ratios very close to 1). GPS performed poorly when the signal was strong but was closer to VIF when the signal was weaker; Lasso was consistently worse than VIF. When the multicollinearity of the variables was moderate (with $\theta = 0.9$), Figure 12 shows that stepwise regression and FoBa could have a >5% gain over the VIF regression; the performance of Lasso remained the same, but the performance of GPS was almost identical to that of VIF regression when the signal was weak. Thus, GPS benefited substantially from its shrinkage in cases with large noise and strong multicollinearity. In a nutshell, the VIF regression maintains its good performance under changing signal-to-noise ratios.

We also compared the Naïve algorithm without the VIF correction under this setup in Figure 13. Its performance was identical to that of VIF regression when $\theta = 0.5$. This performance under weak multicollinearity was guaranteed in the literature (see, e.g., Tropp 2004; Cai and Wang 2010). However, when the multicollinearity was moderate ($\theta = 0.9$), the Naïve algorithm

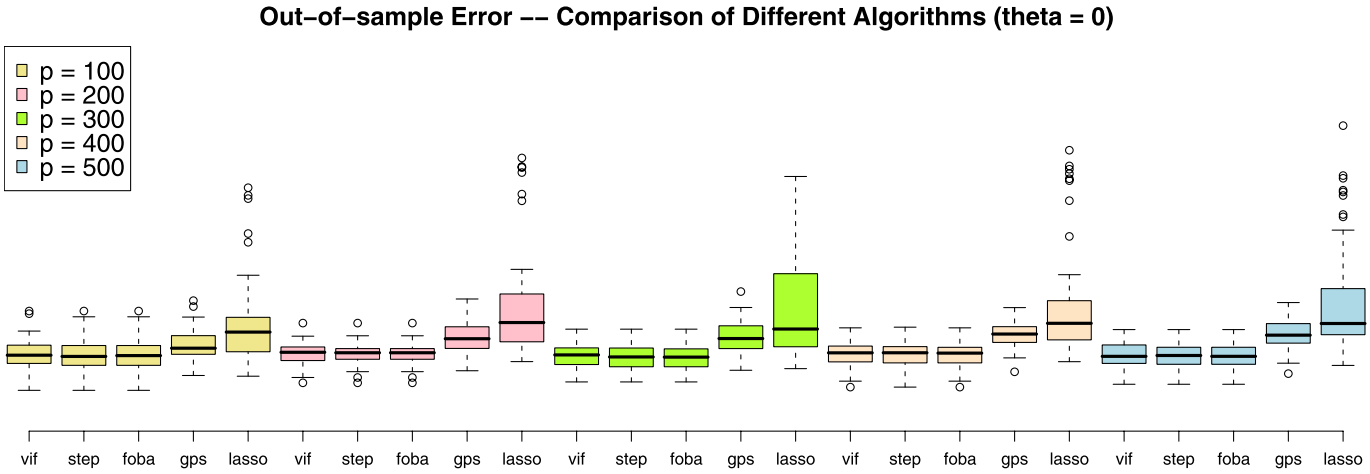

Figure 9. Out-of-sample mean squared errors of the models chosen by the five algorithms in the simulation study with independent distributed features. The number of features p varied from 100 to 500 (from left to right in the figure).

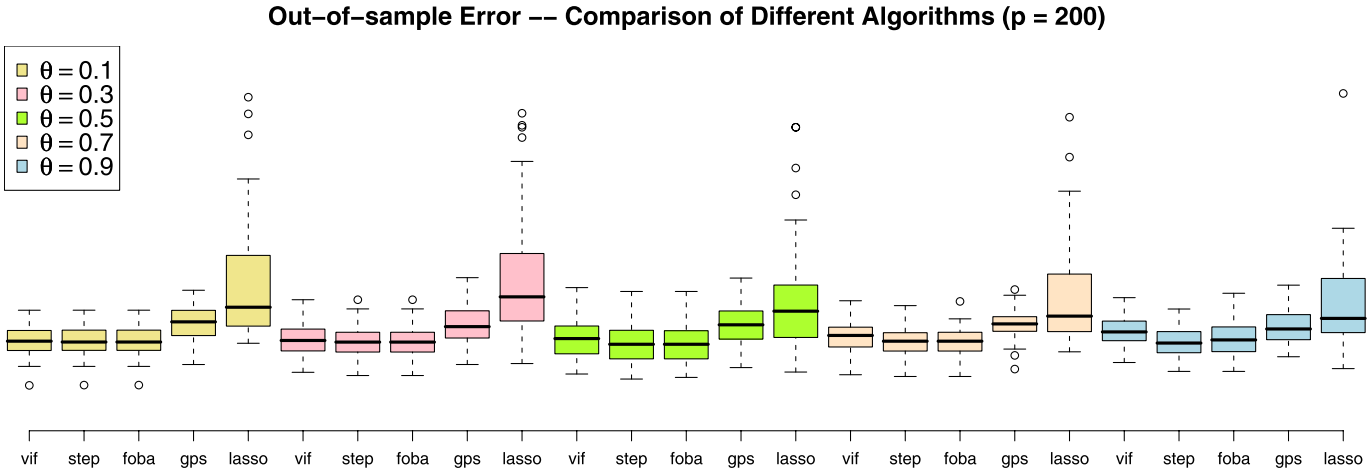

Figure 10. Out-of-sample mean squared errors of the models chosen by the five algorithms. The 200 candidate features were simulated under the second scenario with $\theta = 0.1, 0.3, 0.5, 0.7$, and 0.9 in Σ_2 (from left to right in the figure).

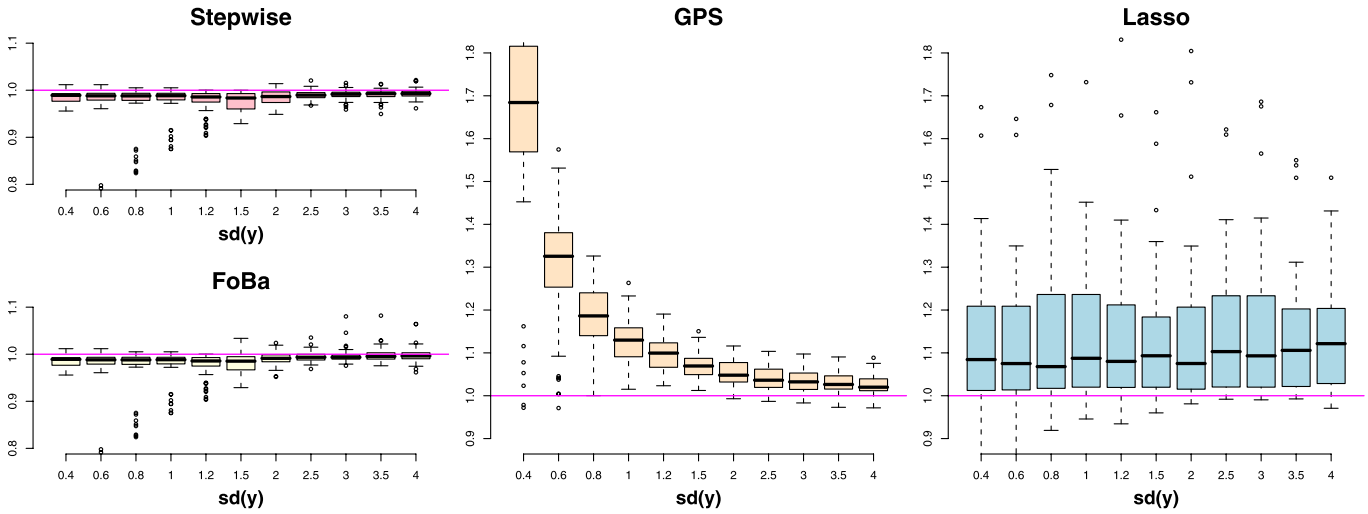

Figure 11. Ratio of out-of-sample mean squared errors of the models chosen by the other four algorithms to that of VIF regression. A ratio less than (greater than) 1 implies a better (worse) performance of the algorithm compared with the VIF regression. The 200 features were simulated under the second scenario with $\theta = 0.5$ in Σ_2. The online version of this figure is in color.

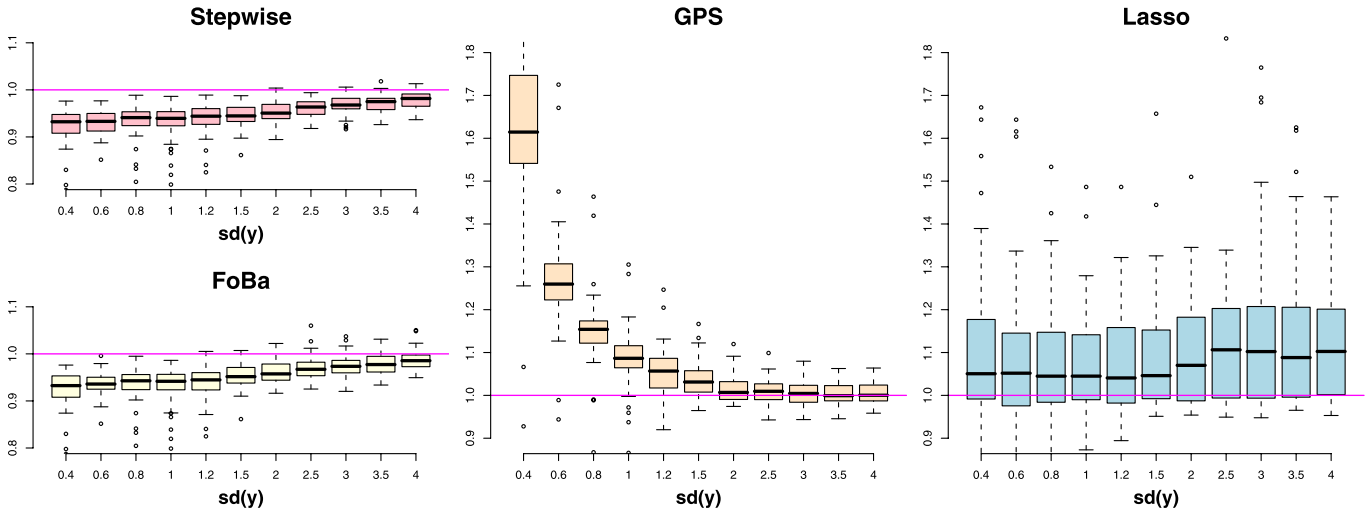

Figure 12. Ratio of out-of-sample mean squared errors of the models chosen by the other four algorithms to that of VIF regression. A ratio less than (greater than) 1 implies a better (worse) performance of the algorithm compared with the VIF regression. The 200 features were simulated under the second scenario with $\theta = 0.9$ in Σ_2. The online version of this figure is in color.

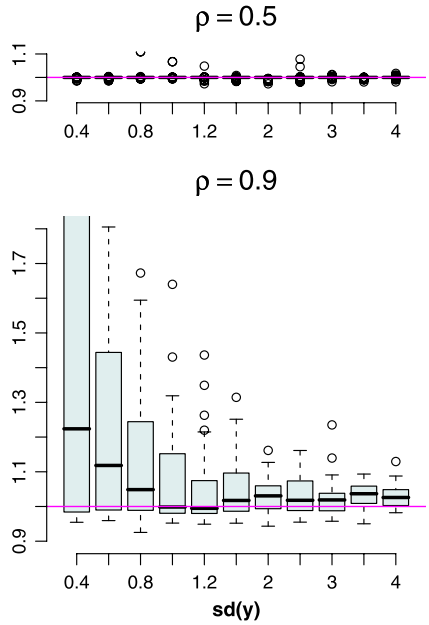

Figure 13. Ratio of out-of-sample mean squared errors of the models chosen by the Naïve algorithms to that of VIF regression. A ratio less than (greater than) 1 implies a better (worse) performance of the algorithm compared with VIF regression. The 200 features were simulated under the second scenario with $\theta = 0.5$ and $\theta = 0.9$ in Σ_2. The online version of this figure is in color.

performed worse than VIF correction, especially when the signal was relatively strong. These results again demonstrate the need for the VIF correction in real applications, where testing the mutual incoherence (weak multicollinearity) is NP hard.

6.6 Robustness of w_0 and Δw

In our algorithm we have two parameters, w_0 and Δw, that represent the initial wealth and the investment. In this section we investigate how the choices of these two parameters might affect the performance of our algorithm.

We took the first simulation scheme and simulated $p = 500$ independent predictors. The response variable $\mathbf{y}$ was generated as the sum of $q = 6$ randomly sampled predictors plus a standard normal noise. We then let the VIF regression algorithm choose models, with w_0 varying from 0.05 to 1 and Δw varying from 0.01 to 1. We computed the out-of-sample mean squared errors for each pair of $(w_0, \Delta w)$. We repeated the whole process 50 times.

Figure 14 illustrates the median, median absolute deviation (MAD), mean, and standard deviation (SD) of these out-of-sample mean squared errors. We note that the robust measures, median and MAD, of these out-of-sample errors were very stable and remained the same for almost all $(w_0, \Delta w)$ pairs. The less robust measures, mean and SD, showed some variation for the pairs with small values. With fixed Δw, the out-of-sample performance did not change much with different w_0's. In fact, because w_0 will be washed out with an exponential decay rate in the number of candidate variables being searched, it matters only for the first few important variables, if there are any.

The out-of-sample mean squared errors with large w_0 and large Δw tended to be small and to have small variance. This is because $o(n/\log n)$ variables can be allowed in the model without overfitting (see, e.g., Greenshtein and Ritov 2004). Thus it will not hurt to include more variables by relaxing w_0 and Δw for prediction purposes. Although the pair that we used for all of the simulations, $w_0 = 0.5$ and $\Delta w = 0.05$, has relatively higher mean squared errors, we are more interested in the statistical ability of better controlling mFDR. The numerical experiments in this section suggest that were prediction accuracy the only concern, then one could use larger w_0 and Δw.

7. REAL DATA

In this section we apply our algorithm to three real data sets: the Boston housing data, a set of personal bankruptcy data, and a call center data set. The Boston housing dataset is small enough to allow us to compare all of the algorithms and show that VIF regression maintains accuracy even with a substantially increased speed. The bankruptcy data are of moderate size (20,000 observations and 439 predictors or, on average, more than 27,000 predictors when interactions are included), but interactions that contribute significantly to the prediction accuracy increase the number of features to the tens of thousands, making the use of much of the standard feature selection and regression software impossible. The call center dataset is even larger, with more than 1 million observations and, once interactions are included, more than 14,000 predictors.

7.1 Boston Housing Data-Revisited

Here we revisit the Boston Housing data discussed in Section 4. Discussions of this dataset in the literature have dealt mainly with 13 variables. To make the problem more demanding, we included multiway interactions up to order 3 as potential variables. This expands the scope of the model and allows a *nonlinear* fit. On the other hand, it produces a feature set with high multicollinearity. We performed five-fold cross-validation on the data; that is, we divided the data into five pieces, built the model based on four of these pieces, and tested the model on the remaining piece. The results are summarized in Table 4. Not surprisingly, stepwise regression gave the best performance overall, because it attempted to build the *sparsest* possible model with strong collective predictability and thus did not suffer much from the multicollinearity. However, the strong multicollinearity caused trouble for GPS, the leader in the case without interactions. A possible explanation for this is that due to the strong collinearity, GPS had a hard time making a unified decision on the working penalty for the different folds. This variability in penalties led to a much larger variance in model performances. As a result, the test errors tended to be large and to have a high variance, as shown in Table 4. The same problem was seen with Lasso, which did well only with small p and weak collinearity. VIF regression did well in both cases, because it attempted to approximate the search path of stepwise regression; the substantially improved speed came at a cost of slightly higher errors.

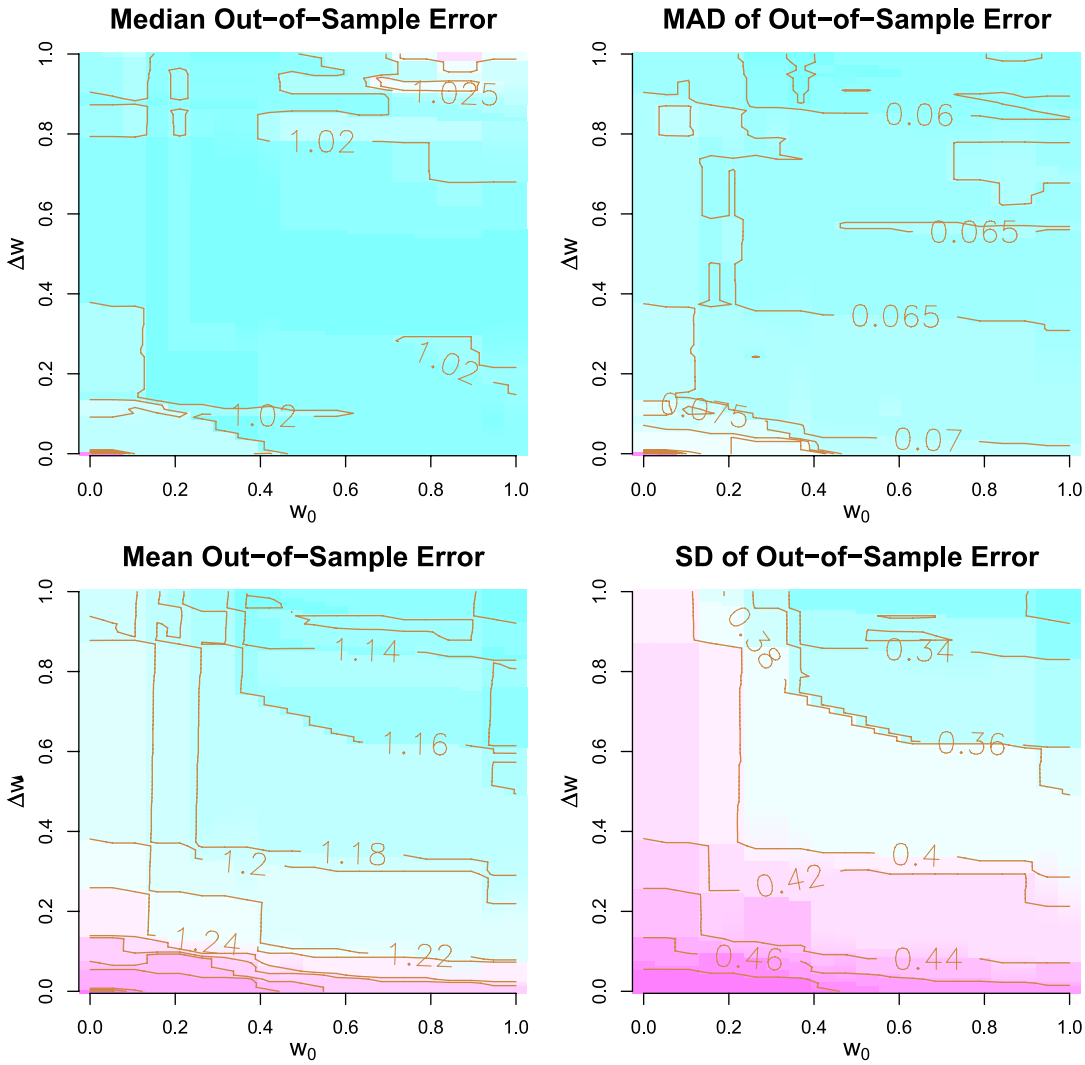

Figure 14. Statistics of out-of-sample mean squared errors with various w_0 and Δw. The online version of this figure is in color.

7.2 Bankruptcy Data

We also applied VIF regression to the bankruptcy data that were originally used by Foster and Stine (2004). This sample dataset contains 20,000 accounts and 147 features, 24 of which are categorical. It has substantial missing data. It is well understood that missing data serve to characterize the individual account behaviors (Jones 1996); that is, knowing which data are missing improves model predictivity. Thus, instead of filling in with expected values based on the observed data, we use an indicator for each of them, as done by Foster and Stine (2004). We also decompose each of the 24 categorical variables that have categories (l) >2 into $l-1$ dummy variables. Thus we have a total of 439 features for our linear model. To dynamically select interaction terms, we first apply VIF regression on the 439 linear features to get a baseline subset, $\mathcal{C}_0$. We then apply VIF regression with subsampling size $m = 400$ on the interaction terms of the selected variables in $\mathcal{C}_0$ and all of the features. Thus, we considered a total of $p = (|\mathcal{C}_0| + 1) \times 439$ candidate variables, as summarized in Table 5.

To evaluate the classification performance, we perform a five-fold cross-validation and use the 0–1 loss function to compute the in-sample and out-of-sample classification errors for each fold. We compared two different cutoff rules, $\xi_1 = 1 - \#\text{BANKRUPTCIES}/n_{\text{CV}}$, where $\#\text{BANKRUPTCIES}$ is the number of bankrupt accounts in sample, and $\xi_2 = 8/9$.

Table 4. Boston housing data: Average out-of-sample mean squared error in a five-fold cross-validation study. The values in parentheses are the standard error of the these average mean squared errors

		Methods				
Cases	p	VIF	Stepwise	FoBa	GPS	Lasso
No interactions	13	35.77 (26.25)	39.37 (26.11)	41.52 (33.33)	**35.26** (19.56)	37.40 (24.67)
3-interactions	403	26.57 (22.68)	**26.39** (18.54)	31.62 (23.94)	95.75 (98.36)	96.76 (47.10)

Table 5. Bankruptcy data: The performance of VIF and stepwise regression on a five-fold cross-validation

Method	#bankruptcies	$\lvert\mathcal{C}_0\rvert$	p	Time	in.err^1	out.err^1	in.err^2	out.err^2
VIF	366	60.8	27,130	88.6	0.020	0.021	0.021	0.021
Stepwise	–	–	22,389	90	0.023	0.023	0.022	0.022

NOTE: Time: CPU running time in minutes. in.err^1/out.err^1: In-sample classification errors/Out-of-sample classification errors using ξ_1. in.err^2/out.err^2: In-sample classification errors/Out-of-sample classification errors using ξ_2. All numbers are averaged over the five folds.

We also performed a comparison with stepwise regression by generating 22,389 predictors and using stepwise regression to choose variables. Given a time limit of 90 minutes, stepwise regression could select (on average) only four variables, compared with 400 features selected by VIF. We were not able to run the other three algorithms on these data.

7.3 Call Center Data

The call center data that we explore in the section are collected by an Israeli bank. On each day, the number of calls to the customer center was counted every 30 seconds. This call value is the dependent variable to be predicted. The data were collected between November 1, 2006, and April 30, 2008, a total of 471 days (a few days are missing). Thus we have a total of $n = 1{,}356{,}480$ observations. Similar data sets have been investigated by Brown et al. (2005) and Weinberg, Brown, and Stroud (2007).

To ensure approximately normal errors, we performed a variance stabilization transformation (Brown et al. 2005) to the number of counts N:

$$\mathbf{y} = \sqrt{N + 1/4}.$$

The variables that we investigated for possible inclusion in the model included *day of week* $\{x_d\}_{d=1}^{6}$, *time of day* ϕ_t^f and ψ_t^f, and *lags* $\mathbf{y}_{t-k}$. For time of day, we considered Fourier transforms

$$\phi_t^f = \sin\left(\frac{2\pi f \cdot t}{\omega}\right) \quad \text{and} \quad \psi_t^f = \cos\left(\frac{2\pi f \cdot t}{\omega}\right),$$

where $\omega = 2880$, the length of the daily period, and f varies from 1 to 2^{10}. We also considered interactions between day of week and time of day, $\{\phi_t^f \cdot x_d\}$ and $\{\psi_t^f \cdot x_d\}$ as explanatory variables. This resulted in a set of 2054 base predictors and 12,288 interactions.

We again performed five-fold cross-validation to test our performance. Our VIF regression selected on average 82 of the features and gave an in-sample R^2 of 0.779 and an out-of-sample R^2 of 0.623. The features selected were primarily interactions between day of week and time of day, as summarized in Table 6.

Note that the in-sample performance is better than the out-of-sample performance because of the autoregressive nature of these data. The feature selection criteria that we used guarantees only that there will be no overfitting for the case of independent observations. For nonindependent observations, the effective sample size is smaller than the actual number of observations, and thus adjusted criteria should be taken into account. We also considered adding autoregressive effects (i.e., lag variables $\{y_{t-k}\}$) in the model. We gained both in-sample and out-of-sample R^2 as high as 0.92. However, in the typical use of models of call center data, estimating the number of calls to determine staffing levels, $\{y_{t-k}\}$, is not available at the time that the staffing decisions need to be made and so cannot be used for prediction. The speed and flexibility of our algorithm enable us to efficiently extract informative relationships for such large-scale data.

8. DISCUSSION

Fast and accurate variable selection is critical for large-scale data mining. Efficiently finding good subsets of predictors from numerous candidates can greatly alleviate the formidable computation task, improve predictive accuracy, and reduce the labor and cost of future data collection and experiments. Among the various variable selection algorithms available, stepwise regression has been empirically shown to be accurate but computationally inefficient; l_1 and l_ε algorithms are less accurate in highly sparse systems. In this article we have proposed a hybrid algorithm, VIF regression, that incorporates a fast and simple evaluation procedure. VIF regression can be adapted to various stepwise-like algorithms, including a streamwise regression algorithm using an α-investing rule. Because of the one-pass nature of the streamwise algorithm, the total computational complexity of this algorithm can be reduced to $O(pn)$ as long as the subsample size is $m = O(n/q^2)$, which can be easily achieved in large-scale datasets. Furthermore, by using an α-investing rule, this algorithm can control mFDR and avoid overfitting. Our experimental results demonstrate that our VIF algorithm is substantially as accurate as, and faster than, other algorithms for large-scale data. Based on these results, we believe that the VIF algorithm can be fruitfully applied to many large-scale problems. VIF regression code in R is available at the CRAN repository (*http://www.r-project.org/*).

[Received February 2010. Revised October 2010.]

Table 6. Call center data: Performance of VIF and selected variables on a five-fold cross-validation

	# of selected variables			Performance	
	Day of week	Time of day	Interactions	In-sample R^2	Out-of-sample R^2
Average	6	18.4	57.8	0.779	0.623

NOTE: All numbers are averaged over the five folds.

REFERENCES

Benjamini, Y., and Hochberg, Y. (1995), "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society, Ser. B*, 57, 289–300. [237]

Breiman, L., and Freedman, D. (1983), "How Many Variables Should Be Entered in a Regression Equation?" *Journal of the American Statistical Association*, 78, 131–136. [232]

Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., and Zhao, L. (2005), "Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective," *Journal of the American Statistical Association*, 100, 36–50. [246]

Cai, T., and Wang, L. (2010), "Orthogonal Matching Pursuit for Sparse Signal Recovery," technical report, University of Pennsylvania. [240,242]

Candes, E. J., and Tao, T. (2007), "The Dantzig Selector: Statistical Estimation When p Is Much Larger Than n," *The Annals of Statistics*, 35, 2313–2351. [232,233]

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression" (with discussion), *The Annals of Statistics*, 32, 407–451. [232, 233,240]

Efron, B., Tibshirani, R., Storey, J. D., and Tusher, V. (2001), "Empirical Bayes Analysis of a Microarray Experiment," *Journal of the American Statistical Association*, 96, 1151–1160. [237]

Foster, D. P., and Stine, R. A. (2004), "Variable Selection in Data Mining: Building a Predictive Model for Bankruptcy," *Journal of the American Statistical Association*, 99, 303–313. [236,245]

——— (2008), "Alpha-Investing: A Procedure for Sequential Control of Expected False Discoveries," *Journal of the Royal Statistical Society, Ser. B*, 70, 429–444. [232,233,236,237,239]

Friedman, J. H. (2008), "Fast Sparse Regression and Classification," technical report, Standford University. [232-234,240]

Friedman, J. H., Hastie, T., and Tibshirani, R. (2010), "Regularization Paths for Generalized Linear Models via Coordinate Descent," *Journal of Statistical Software*, 33, 1–22. [233]

Greenshtein, E., and Ritov, Y. (2004), "Persistency in High Dimensional Linear Predictor-Selection and the Virtue of Over-Parametrization," *Bernoulli*, 10, 971–988. [232,244]

Guyon, I., and Elisseeff, A. (2003), "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, 3, 1157–1182. [232]

Hastie, T., Tibshirani, R., and Friedman, J. (2009), *The Elements of Statistical Learning* (2nd ed.), New York: Springer-Verlag. [233]

Jones, M. P. (1996), "Indicator and Stratification Methods for Missing Explanatory Variables in Multiple Linear Regression," *Journal of the American Statistical Association*, 91, 222–230. [245]

Konishi, S. (1978), "An Approximation to the Distribution of the Sample Correlation Coefficient," *Biometrika*, 65, 654–656. [239]

Lin, D., Foster, D. P., Pitler, E., and Ungar, L. H. (2008), "In Defense of l_0 Regularization," in *ICML/UAI/COLT 2008 Workshop: Sparse Optimization and Variable Selection*, Helsinki, Finland. [233]

Miller, A. (2002), *Subset Selection in Regression* (2nd ed.), Boca Raton, FL: Chapman & Hall. [232,234]

Natarajan, B. K. (1995), "Sparse Approximate Solutions to Linear Systems," *SIAM Journal on Computing*, 24, 227–234. [232,234]

Storey, J. D. (2002), "A Direct Approach to False Discovery Rates," *Journal of the Royal Statistical Society, Ser. B*, 64, 479–498. [237]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society, Ser. B*, 58, 267–288. [240]

Tropp, J. A. (2004), "Greed Is Good: Algorithmic Results for Sparse Approximation," *IEEE Transactions on Information Theory*, 50, 2231–2242. [242]

Wang, H. (2009), "Forward Regression for Ultra-High Dimensional Variable Screening," *Journal of the American Statistical Association*, 104, 1512–1524. [239]

Weinberg, J., Brown, L. D., and Stroud, J. R. (2007), "Bayesian Forecasting of an Inhomogeneous Poisson Process With Applications to Call Center Data," *Journal of the American Statistical Association*, 102, 1185–1198. [246]

Zhang, T. (2009), "Adaptive Forward–Backward Greedy Algorithm for Sparse Learning With Linear Models," in *Advances in Neural Information Processing Systems*, Vol. 21, Cambridge, MA: MIT Press, pp. 1921–1928. [232-234,236,240]

Zhou, J., Foster, D. P., Stine, R. A., and Ungar, L. H. (2006), "Streamwise Feature Selection," *Journal of Machine Learning Research*, 7, 1861–1885. [233,239]