

Shotgun Stochastic Search for “Large p ” Regression

Chris Hans^{*†}, Adrian Dobra^{*‡} and Mike West^{*}

May 2005

Abstract

Model search in regression with very large numbers of candidate predictors raises challenges for both model specification and computation, and standard approaches such as Markov chain Monte Carlo (MCMC) and step-wise methods are often infeasible or ineffective. We describe a novel *shotgun stochastic search* (SSS) approach that explores “interesting” regions of the resulting, very high-dimensional model spaces to quickly identify regions of high posterior probability over models. We describe algorithmic and modeling aspects, priors over the model space that induce sparsity and parsimony over and above the traditional dimension penalization implicit in Bayesian and likelihood analyses, and parallel computation using cluster computers. We discuss an example from gene expression cancer genomics, comparisons with MCMC and other methods, and theoretical and simulation-based aspects of performance characteristics in large-scale regression model search. We also provide software implementing the methods.

Key Words: Model averaging, parallel computing, regression model uncertainty, stochastic search, variable selection.

1 Introduction

Regression variable uncertainty – framed as either model selection or model averaging – raises modeling and computational challenges as the number of candidate predictor variables increases. Standard methods including stepwise methods, leaps-and-bounds and Markov chain Monte Carlo (MCMC) (Furnival and Wilson, 1974; Clyde and George, 2004) can often quickly find “good” models when the number of predictors is relatively small. Stepwise methods are infeasible in higher dimensional problems, are prone to entrapment in local

^{*}Institute of Statistics and Decision Sciences, Duke University, Durham NC 27708

[†]email: hans@isds.duke.edu

[‡]Department of Molecular Genetics & Microbiology, Duke University Medical Center, Durham NC 27710

maxima of model space, and often do not provide an adequate representation of the model space with the increasingly complex patterns of collinearity that are typical with many variables. MCMC algorithms designed to explore the posterior distribution over regression model spaces (e.g., George and McCulloch, 1993, 1997; Green, 1995; Madigan and York, 1995; Geweke, 1996; Raftery et al., 1997; Brown et al., 1998b) rely on Gibbs sampling (Gelfand and Smith, 1990) or Metropolis-Hastings algorithms, but are increasingly ineffective due to slow convergence in higher dimensions. Outside of the regression model context, MCMC approaches have been used for model space exploration by Chipman et al. (1998) for Bayesian CART models, by Wong et al. (2003) for covariance selection models, and by Tadesse et al. (2005) for clustering.

We introduce a novel *shotgun stochastic search* (SSS) method that is inspired by MCMC but offers the ability to much more rapidly identify probable models and swiftly move around in the space of models as dimension escalates. SSS evaluates many, many models guided by model scores that represent the unnormalized posterior probabilities over models. The computational method can be trivially adapted to model search using any other model "score", such as A/BIC, if desired. The approach rapidly identifies and catalogues many models, and is effective at swiftly identifying large numbers of related models that arise as a result of severe collinearity in large problems, such as in applications in gene expression analysis, while also exploring multiple separate regions of model space representing local modes over models. The method is parallelizable, which enables scaling to problems with very large numbers predictors that would otherwise be simply infeasible.

SSS is introduced and developed in Section 2, and its use in linear and binary regression discussed in Section 3. Operating characteristics in certain models are discussed in Section 4, together with comparison with MCMC approaches. We provide an example in gene expression analysis in Section 5, and some concluding comments in Section 6.

2 Searching for Regression Models

2.1 Shotgun Stochastic Search

The example data set described in Section 5 has $p = 4,514$ possible predictor variables. If we consider regression models with up to five predictors, there are over 10^{16} models; even in this constrained model space, enumeration is impossible. Denote the model space by Γ . Stochastic model search aims to discover and evaluate a (large) set of models, Γ^* , to be used in understanding model (variable subset selection) uncertainty, and for prediction. Regression model shotgun stochastic search (RMSSS) is such a method. It is a regression model specific implementation of a general class of *shotgun stochastic search* (SSS) methods. SSS is an iterative, local-move, neighborhood-based procedure in which we:

STEP 1 Use the “current” model to define a neighborhood of proposal models;

STEP 2 Evaluate each proposal model in this neighborhood *in parallel*; and then

STEP 3 Choose a new current model from the proposals.

A key idea is that, for any “current” model, there may be many other models with similar “fit” to the data – models with over-lapping or collinear predictors. Quickly identifying and evaluating these models provides a rich description of part of the model space and a new set of competitive models from which to choose the next move. This generates multiple candidate models and “shoots out” proposed moves in various directions in model space, hence the “shotgun” terminology.

The neighborhood of the current model must be comprehensive enough to allow the search to move easily throughout the model space. This is accomplished by considering each possible predictor variable in one of the proposal models at each iteration. This approach has the added benefit that, over the course of the search, every candidate variable is evaluated in the context of many, many different regression models. Critically, **STEP 2** can be parallelized: each of the proposal models can be evaluated independently on sepa-

rate processors, providing a clear advantage of SSS procedures over MCMC algorithms in which models are proposed and evaluated one-at-a-time, sequentially. The criterion used to compare models is problem-specific; in a Bayesian analysis, as described here, the model “score” is posterior probability, whereas the search method can be applied with other notions of model fit/score.

2.2 Regression Model SSS

The two major components of SSS are the choice of neighborhood (how to “shoot out” proposal models) and the model move/sampling strategy. The neighborhood component should allow us to consider each of the possible predictor variables at each step, and permit regression models of various dimensions to allow the search to move freely across model size. We take the neighborhood to be every regression model that is a one-variable change to the current model.

Let p be the total number of possible predictor variables, and γ be a $p \times 1$ indicator vector with $\gamma_j = 1(0)$ if variable j is in the regression model (or not). For a current regression model of dimension k (i.e., having k predictor variables) the neighborhood has three elements, $\text{nbrd}(\gamma) = \{\gamma^+, \gamma^\circ, \gamma^-\}$, where γ^+ is a set containing neighboring models of dimension $k + 1$, called the “addition” moves, γ° is a set containing neighboring models of dimension k , called the “replacement” moves, and γ^- is a set containing neighboring models of dimension $k - 1$, called the “deletion” moves. Set γ^+ contains all of the models obtained by adding any one of the $p - k$ remaining predictor variables; γ^- is the k models obtained by deleting any one current variable; γ° is the set obtained by replacing any one current variable with any one of the $p - k$ remaining.

For example, with $p = 5$, if the current regression model is $\{x_1, x_3, x_4\}$, corresponding to $\gamma = (1, 0, 1, 1, 0)'$, then

$$\gamma^- = \left\{ \{x_3, x_4\}, \{x_1, x_4\}, \{x_1, x_3\} \right\},$$

$$\begin{aligned}\gamma^\circ &= \bigcup_{j \in \{2,5\}} \left\{ \{x_1, x_3, x_j\}, \{x_1, x_j, x_4\}, \{x_j, x_3, x_4\} \right\}, \\ \gamma^+ &= \bigcup_{j \in \{2,5\}} \{x_1, x_3, x_4, x_j\}.\end{aligned}$$

Note that when $2 \leq k < p$, $|\gamma^+| = p - k$, $|\gamma^\circ| = k(p - k)$ and $|\gamma^-| = k$, with the convention that $\gamma^+ = \emptyset$ when $k = p$. We evaluate the null model, $\gamma = \mathbf{0}$, and all possible one variable models before starting the search and hence only allow SSS to consider models of at least dimension $k = 2$.

As p is typically large, we have $|\gamma^\circ| \gg |\gamma^+| \gg |\gamma^-|$, which can be problematic for sampling. If all of the models were to have equal weight and we sampled one model directly from $\text{nb}d(\gamma)$, then as $p \rightarrow \infty$ the probability of staying in the same dimension goes to $k/(k + 1)$, the probability of increasing goes to $1/(k + 1)$ and the probability of decreasing dimension goes to zero. To move across dimension effectively, we break sampling into two parts: three models, γ_*^+ , γ_*° and γ_*^- are sampled from γ^+ , γ° and γ^- , respectively, and then one of the three selected.

The (unnormalized) posterior probability, $p(\gamma|y) \propto p(y|\gamma)p(\gamma)$, is evaluated for each model generated in SSS. BIC can be viewed as an approximation to the marginal likelihood of a give model, $p(y|\gamma)$, under a reference prior distribution (Raftery, 1995) and so could be used in similar fashion. Other scores such as R^2 and AIC can be used, but the user would have to decide how to use these scores to move from model to model across iterations, i.e. how to normalize the scores into a probability vector from which to sample. In general we will refer to a score for a model γ that can be normalized within a set of scores to become a probability as $S(\gamma)$.

Regression Model Shotgun Stochastic Search Schema

Let γ be a regression model and let $S(\gamma)$ be its corresponding (unnormalized) score. Initialize an empty model list, Γ^ , that will contain the best B regression models evaluated.*

Given a starting model $\gamma^{[0]}$, iterate in $t = 1, \dots, T$ the following steps:

STEP 1 In parallel, compute $S(\gamma)$ for all $\gamma \in \text{nb}d(\gamma^{[t]})$, constructing γ^+ , γ° and γ^- . Update the list of the overall best models evaluated, Γ^* .

STEP 2 Sample γ_*^+ , γ_*° and γ_*^- , from γ^+ , γ° and γ^- , respectively, with probabilities proportional to $S(\gamma)^{\alpha_1}$, normalized within each set.

STEP 3 Sample $\gamma^{[t+1]}$ from $\{\gamma_*^+, \gamma_*^\circ, \gamma_*^-\}$ with probability proportional to $S(\gamma)^{\alpha_2}$, normalized within this set.

The positive ‘‘annealing’’ parameters α_1 and α_2 control how greedy the search is: values less than one flatten out the proposal distribution, allowing the search to wander around more freely, whereas very high values lead to a hill-climbing search. Separate values of α_1 could be used for each of the sets γ^+ , γ° and γ^- . In our examples, $\alpha_1 = \alpha_2 = 1$.

As the search progresses, we update the list Γ^* of the highest scoring B models based on every model evaluated in STEP 1 and not solely based on the models sampled in STEP 3, so defining a list representative of the areas of the model space explored.

3 Linear and Binary Regression

3.1 Normal Linear Regression

Consider the normal linear regression model $\mathbf{Y} \sim \mathbf{N}(X\boldsymbol{\beta}, \sigma^2 I_n)$, where \mathbf{Y} is an $n \times 1$ response variable, $X = (x_1, \dots, x_n)'$ is an $n \times p$ design matrix for the n samples, and the x_i are $p \times 1$ vectors of covariate information. Assume that both x and y data are standardized so that we do not include an intercept term.

We assume priors on $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma^2)'$ that are consistent across models in the sense that they are derived from an encompassing model via conditioning. This follows Dobra et al. (2004) in assuming each observation $(y_i, x_i)'$ to have joint normality, $\mathbf{N}(0, \Sigma)$, with $(p + 1) \times (p + 1)$ covariance matrix Σ and a corresponding precision matrix $\Omega = \Sigma^{-1}$. Any given

regression model γ then arises from the conditional distribution $p(y|x, \Sigma)$ where zeros have been placed in the first row and column of Ω in the locations corresponding to zeros in γ . The prior in the regression arises from the prior on Σ . Using an inverse Wishart prior distribution, $\Sigma \sim \text{IW}(\delta, \tau I)$ with δ degrees of freedom and scale matrix τI , the following prior distributions are implied for a regression with k predictor variables:

$$[\beta_\gamma | \sigma^2, \gamma] = \text{N}(0, \tau^{-1} \sigma^2 I_k) \quad \text{and} \quad [\sigma^2 | \gamma] = \text{IG} \left(\frac{\delta + k}{2}, \frac{\tau}{2} \right),$$

where β_γ are the regression coefficients under model γ . As shown in Dobra et al. (2004) this implies posterior distributions

$$[\beta_\gamma | \sigma^2, y, X, \gamma] = \text{N} \left(M_\gamma^{-1} X_\gamma' y, \sigma^2 M_\gamma^{-1} \right), \quad (1)$$

$$[\sigma^2 | y, X, \gamma] = \text{IG} \left(\frac{\delta + k + n}{2}, \frac{\tau + q_\gamma}{2} \right), \quad (2)$$

where X_γ is the design matrix, $M_\gamma = \tau I_k + X_\gamma' X_\gamma$ and $q_\gamma = y' y - y' X_\gamma M_\gamma^{-1} X_\gamma' y$.

In order to compute $p(\gamma|y)$, we need to first compute the marginal (or integrated) likelihood of the data given the model, $p(y|\gamma) = \int p(y|\theta, \gamma) p(\theta|\gamma) d\theta$. Under the above formulation, we can find this quantity in closed form:

$$p(y|\gamma) = \frac{\Gamma \left(\frac{n+\delta+k}{2} \right)}{\pi^{n/2} \tau^{(n-k)/2} |M_\gamma|^{1/2} \{1 + q_\gamma/\tau\}^{(n+\delta+k)/2} \Gamma \left(\frac{\delta+k}{2} \right)}. \quad (3)$$

By Bayes theorem, the posterior probability of any model is $p(\gamma|y) \propto p(y|\gamma)p(\gamma)$. The posterior distributions (1) and (2), along with the marginal likelihood (3), will be used in Section 3.4 in forming model averaged predictions and estimates of variable importance.

3.2 Binary Regression

In the case of independent binary outcomes, y_i , consider the logistic regression $p(y|\beta, \gamma) = \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{1-y_i}$, where $p_i = 1/(1 + \exp\{-(\beta_0 + \mathbf{x}_i' \beta_\gamma)\})$ and \mathbf{x}_i contains only those variables indicated by the model γ . Notice the inclusion of the intercept term β_0 , which is necessary to account for the baseline response probability. Parallel to the linear model case,

take the prior

$$[\beta_0, \boldsymbol{\beta}_\gamma | \boldsymbol{\gamma}] = \mathbf{N}(0, \tau I_{k+1}),$$

where $k = \sum_{j=1}^p \gamma_j$. Again assuming standardized predictor variables, we typically take $\tau = 1$ to place appropriate prior mass on reasonable values of the regression coefficients.

The marginal likelihood, $p(y|\boldsymbol{\gamma})$, is not available in closed form but can be approximated via the Laplace approximation $\hat{p}(y|\boldsymbol{\gamma}) = (2\pi)^{p/2} |\hat{\Sigma}|^{1/2} h(\hat{\boldsymbol{\beta}}|\boldsymbol{\gamma})$ (DiCiccio et al., 1997), where $h(\boldsymbol{\beta}|\boldsymbol{\gamma}) = p(y|\boldsymbol{\beta}, \boldsymbol{\gamma})p(\boldsymbol{\beta}|\boldsymbol{\gamma})$, and

$$\hat{\Sigma} = - \left(\frac{\partial^2 \log h(\hat{\boldsymbol{\beta}}|\boldsymbol{\gamma})}{\partial \hat{\theta}_i \partial \hat{\theta}_j} \right)^{-1}.$$

We find $\hat{\boldsymbol{\beta}} = \arg \max_{\boldsymbol{\beta}} p(y|\boldsymbol{\beta}, \boldsymbol{\gamma})p(\boldsymbol{\beta}|\boldsymbol{\gamma})$, the maximum *a posteriori* estimate of $\boldsymbol{\beta}$, via Newton's method.

3.3 Prior over the Model Space

As dimension increases it is critical to use priors over model dimension that encourage sparsity, as large models are often less interpretable and there is a risk of over-fitting when n is small relative to p . Here we use the simple, standard model selection prior

$$p(\boldsymbol{\gamma}) = \pi^k (1 - \pi)^{p-k}, \tag{4}$$

where $k = \sum_{i=1}^p \gamma_i$ is the number of variables in the model and π is a hyperparameter representing the probability that a variable is in the model (with all variables treated exchangeably). This prior induces a binomial prior distribution over model size, and thus the prior expected model size is $p\pi$. We typically set $\pi = m/p$, with m small relative to p , to encourage sparsity.

3.4 Variable Identification and Inference

Define the model score as the unnormalized posterior probability for a given model, $S(\boldsymbol{\gamma}) = p(y|\boldsymbol{\gamma})p(\boldsymbol{\gamma})$ with $p(\boldsymbol{\gamma}|y) \propto S(\boldsymbol{\gamma})$. In the "top model list" Γ^* , measure the relative importance

of predictor variable x_j by computing

$$\tilde{p}(\gamma_j = 1|y) = C^{-1} \sum_{\gamma \in \Gamma^*} \mathbf{1}_{\{\gamma_j=1\}} p(y|\gamma)p(\gamma), \quad (5)$$

where the normalizing constant is the posterior mass contained in Γ^* , $C = \sum_{\gamma \in \Gamma^*} p(y|\gamma)p(\gamma)$.

If we could have explored the entire space (so that $\Gamma^* = \Gamma$), then (5) would represent the posterior probability of variable inclusion for variable x_j . Rather, as we have only explored some part of the model space, (5) represents the posterior probability of variable inclusion for variable x_j *conditioned* on the set Γ^* . View $\tilde{p}(\gamma_j = 1|y)$ as a measure of the relative importance of variable x_j in the context of the top predictive models.

Similarly, measure the relative importance of individual models by

$$\tilde{p}(\gamma|y) = C^{-1} p(y|\gamma)p(\gamma). \quad (6)$$

Inferences and predictions can now be based on the set of top models assumed to represent a conditional posterior over models.

4 The Nature and Effectiveness of SSS

We report on some evaluation of operating characteristics of SSS from two perspectives. First, assuming the existence of a “true” model $\gamma^* \in \Gamma$, we investigate the ability of SSS to quickly find this model under a variety of conditions. Second, we consider how SSS is related to MCMC algorithms and how it dominates in the ability to rapidly explore highly scoring regions of model space.

4.1 Random Walk SSS

Consider a *fixed dimensional* SSS, one where we condition on a particular number of variables k , assume that the true model γ^* is of dimension k , and only allow moves within this dimension, effectively setting $\text{nbd}(\gamma) = \gamma^\circ$. A fixed dimensional SSS creates a Markov

chain $\{\gamma_t\}$ over the state space of models restricted to size k , $\Gamma^{(k)}$, which contains $\binom{p}{k}$ elements. As we have conditioned on a particular model size, k , we can categorize any model γ as belonging to one of $k + 1$ classes: the class where γ shares *none* of the same variables as the true model, γ^* , the class where γ shares *one* of the same variables as γ^* , up to the class where γ contains *all* k of the same variables as γ^* . Thus we can define the map $\psi(\gamma_t) = Z_t$, where $Z_t \in \{0, \dots, k\}$ and indicates how many of the variables in γ_t are shared by γ^* . As we are only interested the expected time to find the true model, we analyze the induced chain $\{Z_t\}$ which is defined on a much smaller state space. We now are concerned with the problem of finding the expected time for the chain $\{Z_t\}$ to reach state k .

Letting $T(p, k) = \min\{t \geq 0; Z_t = k\}$ be a random variable representing the time to reach the true model when there are p possible predictors and the true model is of dimension k , define the quantities $v_i(p, k) = \mathbb{E}[T(p, k) | Z_0 = i]$, $i = 0, \dots, k$, noting that $v_k(p, k) = 0$. As a technical note, the state space for the chain is $\{\max\{0, 2k - p\}, \dots, k\}$, which is $\{0, \dots, k\}$ for values of p and k that we are interested in ($p \gg k$). The reduced state space in certain situations is due to the fact that, if you have say $p = 4$ and $k = 3$, there are no models with zero variables in common with γ^* . For simplicity of presentation we only consider cases here where $2k - p \leq 0$ so that $v_0(p, k)$ is meaningful.

To analyze the chain we must specify the transition matrix $\mathbf{P}_{p,k}$ for a case with p predictor variables conditioned on the true model being of size k . $\mathbf{P}_{p,k}$ is hence a $(k + 1) \times (k + 1)$ stochastic matrix with entries $\mathbf{P}_{p,k}(i + 1, j + 1) = \Pr(Z_{t+1} = j | Z_t = i)$ for $i, j = 0, \dots, k$. The state $k + 1$ is treated as an absorbing state, implying $\mathbf{P}_{p,k}(k + 1, k + 1) = 1$ and $\mathbf{P}_{p,k}(k + 1, l) = 0$ for all $l \neq k + 1$. Once we have specified the other transition probabilities, we can construct $\mathbf{Q}_{p,k}$, a substochastic matrix associated with $\mathbf{P}_{p,k}$, where the (absorbing) row and column of $\mathbf{P}_{p,k}$ have been removed. The vector of expected times to find γ^* is $\mathbf{v}(p, k) = (\mathbf{I}_k - \mathbf{Q}_{p,k})^{-1} \mathbf{1}$. Focus is on $v_0(p, k)$, the expected time to find γ^* starting from a model with no variables in common with γ^* , as randomly choosing a starting point will put

us in this situation with high probability.

As a baseline, first consider a *random walk shotgun stochastic search* (RWSSS). This is a SSS where we set $S(\gamma) = |\text{nbd}(\gamma)|^{-1}$, i.e. where we sample uniformly from the neighborhood around the current model. From this we can specify the elements of $\mathbf{P}_{p,k}$:

$$\Pr(Z_{t+1} = j | Z_t = i) = \begin{cases} \frac{i(p-2k+i)}{k(p-k)} & \text{if } j = i - 1, 0 < i < k, \\ \frac{(k-i)(p-2(k-i))}{k(p-k)} & \text{if } j = i \neq k, \\ \frac{(k-i)^2}{k(p-k)} & \text{if } j = i + 1, i < k, \\ 1 & \text{if } j = i = k, \\ 0 & \text{o.w.} \end{cases} \quad (7)$$

Values of $v_0(p, k)$ appear in Figure 1 for the RWSSS. For $p = 500$, γ^* is found on average in about 125,000 steps when $k = 2$; the number increases to about 20 trillion steps at $k = 6$. Including distinguishing information about the models – by sampling based on their relative posterior probabilities – will reduce these expected times dramatically.

4.2 SSS for Orthogonal Designs

In the special case of an orthogonal design, when $x'_i x_j = 0$ for all $i \neq j$, we can extend the results of the fixed dimensional RWSSS in Section 4.1 to the case of a fixed dimensional SSS. Consider two models of size k , γ_a and γ_b , that differ by only one variable. Let $X_a = (X_1 \ X_2)$ and $X_b = (X_1 \ X_3)$ be the the design matrices for these two models, where X_1 is a set of $k - 1$ common variables. Under the model specified in Section 3.1, the ratio of marginal likelihoods for these two models under orthogonality is

$$\frac{p(y|\gamma_a)}{p(y|\gamma_b)} = \left(\frac{a - y'X_1X_1'y - y'X_3X_3'y}{a - y'X_1X_1'y - y'X_2X_2'y} \right)^\nu,$$

where $\nu = (n + \delta + k)/2$ and $a = (\tau + n - 1)^2$. The numerator and denominator differ only by the last term, a scaled version of the least squares estimate $\hat{\beta}_j = (n - 1)^{-1}x'_j y$, which should be related to whether or not the corresponding variables are shared by the true model: $\hat{\beta}_j$ should be relatively large if x_j is in the true model and relatively small otherwise.

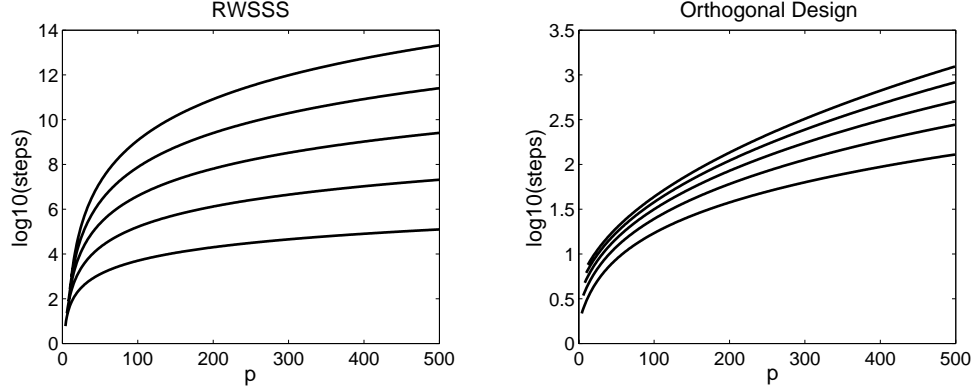


Figure 1: Expected steps (log base 10 scale) to find the true model for RWSSS and SSS under an orthogonal design for various values of p and k . The five lines represent $k = 2, \dots, 6$, with the lowest line being $k = 2$ in each plot. Details are given in Sections 4.1 and 4.2.

As above, consider a Markov chain $\{Z_t\}$ on the state space $\{0, \dots, k\}$. To compute the transition probabilities under an orthogonal design, we make the simplifying assumption that all variables that are not in the true model have the same (relatively small) scaled regression coefficient $\epsilon = x'_j y$, and that all of the variables that are in the true model have the same (relatively large) scaled regression coefficient $\lambda = x'_j y$. After specifying these two values we can compute the transition probabilities $P_{p,k}(i, j)$ (see Appendix A).

The second panel of Figure 1 shows expected hitting times as a function of p for values $k = 2, \dots, 6$ under an orthogonal design. Here we took $n = 500$ and set $\epsilon = (n - 1)0.005$, $\lambda = (n - 1)0.1$, $\tau = 1$ and $\delta = 3$. As seen in Figure 1, under an orthogonal design the number of steps required to hit the true model is drastically smaller under SSS than RWSSS: the expected time to find the true model for model spaces with p around 500 is on the order of several thousand steps. We note that this is the expected time for the chain to achieve $Z_t = k$, however in SSS the true model would be evaluated the step after the chain achieved $Z_t = k - 1$, because one of the models in $\text{nbnd}(\gamma_t) = \gamma^\circ$ would be the true model. Hence we could take state $k - 1$ as the absorbing state and find the expected time until the true model is *evaluated*, which of course will be smaller.

4.3 Simulated Data Example

In this section we report a simulation study based on a real dataset to demonstrate the effectiveness of SSS as the number of possible predictor variables, p , increases. Here we do not restrict ourselves to a fixed dimensional SSS as was considered in the previous two sections, instead we allow SSS to move across dimension as described in Section 2.2. The data on which we base the simulation is a gene expression dataset from a survival study in brain cancer based at the W.M. Keck Center for Neuro-Oncology at Duke University. A detailed description of the data, along with an initial analysis, can be found in Rich et al. (2005).

The study consists of $n = 41$ patients, and for each patient we have gene expression data consisting of $p = 8,408$ genes from a tumor specimen. We selected four genes from the dataset as the variables comprising the “true” model γ^* and simulated $m = 1, \dots, 50$ outcomes using the actual gene expression values x_{ij} for the $j = 1, \dots, 4$ “true” variables according to the regression model

$$y_i^{(m)} = 1.3x_{i1} + 0.3x_{i2} - 1.2x_{i3} - 0.5x_{i4} + \varepsilon_i^{(m)}, \quad (8)$$

for $i = 1, \dots, 41$ where the $\varepsilon_i^{(m)}$ are i.i.d. mean zero normal random variables with variance 0.5. The simulated outcomes were then standardized to have mean zero and unit variance within each of the 50 simulations.

To assess the performance of SSS as the size of the dataset increases, we ran SSS for the 50 simulated responses using datasets with increasing values of p , as shown in Table 1. The datasets were constructed by first reordering the observed $41 \times 8,408$ data matrix X so that the four variables used in the simulation are labeled as variables 1, 2, 3 and 4. To construct a data matrix $X^{(m,p)}$ for a particular simulation, when $p \leq 8,408$ we extracted the first p columns of X to form $X^{(m,p)}$ and then randomly permuted the columns. Hence all 50 datasets $X^{(m,p)}$ for a given $p < 8,408$ contain the same variables and differ only by a column permutation. For the datasets with $p > 8,408$, before permuting the columns we

added $p - 8,408$ columns of random draws from a $N(0, I_{41})$ distribution (after centering and scaling the random draws), effectively adding random noise to the dataset. Note that, for a given $p > 8,408$, different random draws are used for each of the 50 simulated $X^{(m,p)}$.

Prior distributions over the parameter space in the simulation study are consistent with those used in the analysis by Rich et al. (2005), with $\tau = 1$ and $\delta = 3$ as described in Section 3.1. For the model space prior, we set $\pi = 4/p$ as in Section 3.3 in order to maintain focus on sparse models as p increases.

For a given run of SSS, we declared that SSS had found the true model when the true model was evaluated by SSS, i.e. when $\gamma^* \in \text{nb}d(\gamma^{[t]})$. For each value (m, p) , if SSS found the true model within 10,000 iterations we recorded the number of iterations required to find the model and the elapsed time. If the model was not found within 10,000 iterations we recorded the time required for the 10,000 iterations.

Computation was done using 21 processing elements (one master node and 20 compute nodes) on a cluster of dual-processing, 3.1 GHz Intel x86 based machines running Linux. SSS was run for one value of (m, p) at a time using the 21 processors, and the resulting run-time for the simulation was less than ten days.

Results from the simulation study are shown in Table 1 and Figure 2. Overall, SSS found over 96% of the models, as seen in the column labeled “missed” in Table 1, which counts the number of models not found within 10,000 iterations. Additionally, SSS found over 94% of models for datasets with $p \geq 5,000$. While increasing the number of irrelevant variables in the dataset resulted in an increase in the number of iterations needed to find the true model, the true model was still found by SSS a large percentage of the time.

We can use the simulation study to obtain an estimate of the run-time of SSS as a function of p for datasets with a similar number of observations, n . The top panel of Figure 3 displays the run times for datasets with $p \geq 1000$. The lines connect output from SSS runs with the same value of p . A straight line would indicate that the load on the computer cluster was

p	missed	iterations			seconds		
		average	min	max	average	min	max
10		3.38	2	5	0.02	0	1
100		6.66	4	23	0.1	0	1
500		29.68	4	412	1.5	0	22
1000		65.88	3	646	6.64	0	66
2500		181.84	4	1877	46.26	1	492
5000	1	846.88	5	7925	437.02	3	3997
7500	4	790.26	4	9269	601.41	3	7031
8408	1	1001.33	4	8309	878.02	4	7466
10000	3	1542.87	5	9706	1599.51	4	10553
12500	2	1228.35	4	9575	1528.50	5	11937
15000	5	973.07	5	6044	1454.69	7	9403
17500	3	1030.32	5	6938	1785.17	10	12236
20000	3	1259.04	4	9791	2578.11	7	19609

Table 1: Results from the simulation study described in Section 4.3.

constant over runs for a given p ; as the lines are fairly straight we are confident that our results have not been greatly affected by other processes running on the cluster. The first plot in the bottom panel shows the estimated number of iterations per second as a function of p , where the point estimates are the slope coefficients from linear regressions of iterations on seconds for each of the empirical lines shown in the top panel. The second plot on the bottom panel is the inverse of the first. It appears as though for each additional 1000 variables we add to the dataset, SSS takes an extra 0.1 seconds per iteration to run for this example.

4.4 Relationship to MCMC

In cases of high dimensional parameter spaces, MCMC approaches are often used not with the aim of performing Monte Carlo integration to summarize the posterior distribution but rather as a stochastic search tool to identify regions of high posterior probability (or in the context of model selection, to identify the “best” models). In these cases, the Markov chains created by the MCMC algorithms are not expected to converge to a stationary distribution in a reasonable amount of time, rather they are expected to hone in on high posterior regions.

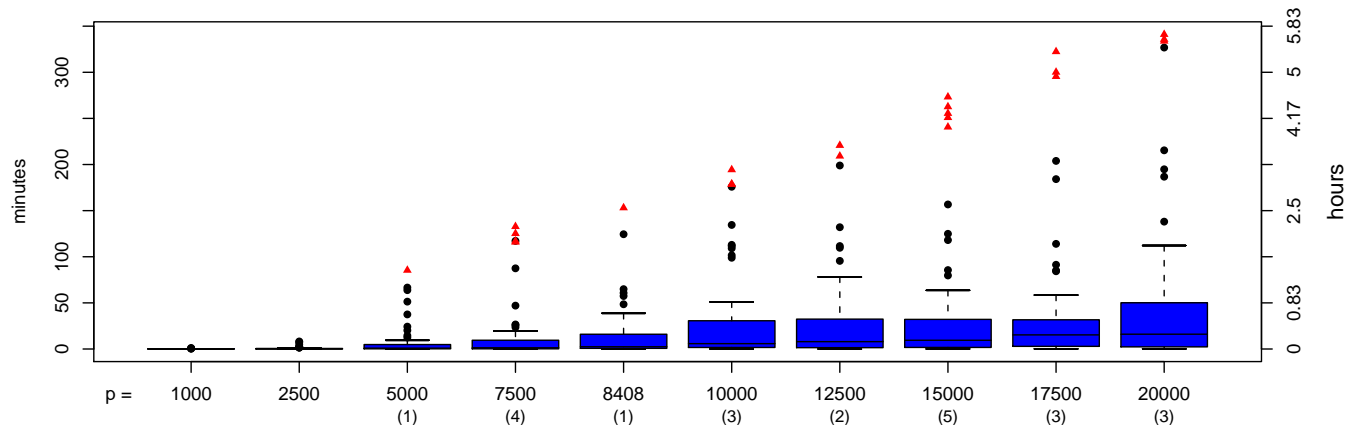


Figure 2: The time required to find the true model for the simulation study described in Section 4.3. The numbers in parentheses indicate the number of models not found by SSS in 10,000 iterations for a given dataset size p , and these runs are plotted as red triangles. The boxplots are based only on runs for which SSS found the true model.

In this section we show that small changes to SSS result in an MCMC algorithm whose particular form has advantages over common MCMC approaches.

Consider use of a Metropolis-Hastings algorithm to sample from a discrete distribution, $P(x)$, where we can evaluate $P(x)$ up to a normalizing constant, $P(x) = Q(x)/Z$. Consider proposal distributions that sample from $P(x)$ restricted to a neighborhood $B(\cdot)$:

$$\begin{aligned} T(x_{t+1}; x_t) &= \frac{P(x_{t+1})\mathbf{1}(x_{t+1} \in B(x_t))}{\sum_{s \in B(x_t)} P(s)} \\ &= \frac{Q(x_{t+1})\mathbf{1}(x_{t+1} \in B(x_t))}{\sum_{s \in B(x_t)} Q(s)}. \end{aligned}$$

As long as we start the chain in a region of nonzero probability, the acceptance probability at each iteration is

$$\alpha = \min \left\{ 1, \frac{Q(B(x_t))}{Q(B(x_{t+1}))} \right\}. \quad (9)$$

We can easily adapt the SSS algorithm described in Section 2.2 to become a Metropolis-Hastings algorithm using the proposal distribution described above. Relating notation, we have $P(x_t)$ is $p(\gamma^{[t]}|y)$, $Q(x_t)$ is $S(\gamma^{[t]}) = p(y|\gamma^{[t]})p(\gamma^{[t]})$, and $B(x_t)$ is $\text{nbd}(\gamma^{[t]})$. After

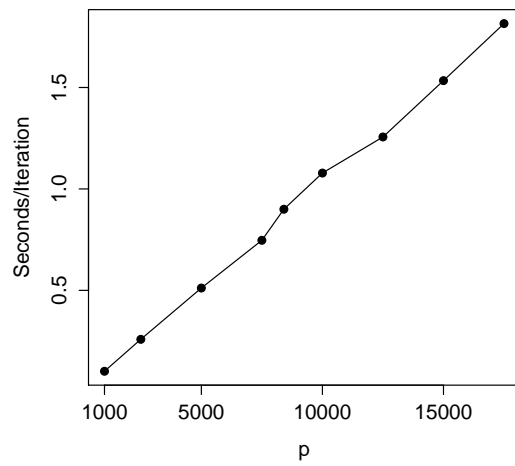
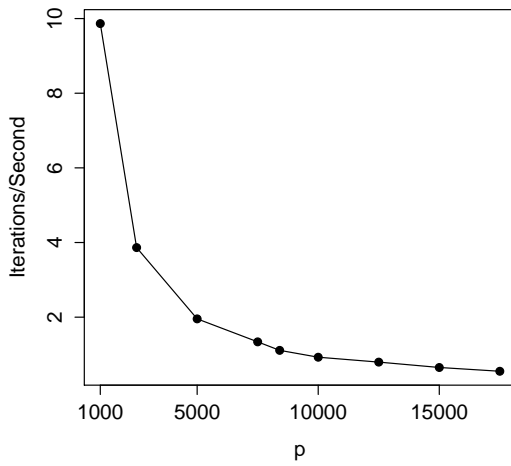
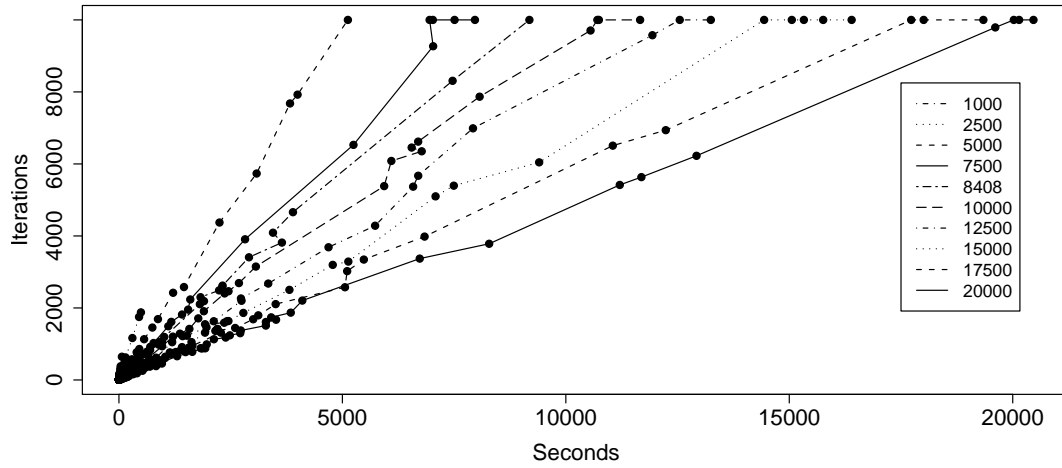


Figure 3: Run times for SSS from the simulation study in Section 4.3. Runs for which the number of iterations equals 10,000 represent runs where γ^* was not found.

performing $S_{\text{STEP 1}}$ at iteration t in SSS, sample a proposal γ' from the discrete distribution $S(\cdot)$ normalized within $\text{nb}d(\gamma^{[t]})$, and set $\gamma^{[t+1]} = \gamma'$ with probability α from (9) (otherwise, set $\gamma^{[t+1]} = \gamma^{[t]}$). $S_{\text{STEP 2}}$ and $S_{\text{STEP 3}}$, which are related to the two stage sampling process that corrects the dimension imbalance, are ignored.

The form of the acceptance probability (9) indicates that MCMC based on SSS behaves differently than MCMC approaches such as the Markov chain Monte Carlo Model Composition (MC^3) algorithm of Madigan and York (1995) and Raftery et al. (1997), and the related approach of Brown et al. (1998a). MC^3 constructs a Markov chain over the model space by first defining a neighborhood $\text{nb}d_*(\gamma) = \gamma^+ \cup \gamma^- \cup \gamma$, using our notation from Section 2.2. A proposal distribution T_* is then defined by setting $T_*(\gamma'; \gamma^{[t]}) = 0$ for all $\gamma' \notin \text{nb}d_*(\gamma^{[t]})$, and setting $T_*(\gamma'; \gamma^{[t]})$ constant for all $\gamma' \in \text{nb}d(\gamma^{[t]})$. As the MC^3 algorithm proceeds, if the chain is in state $\gamma^{[t]}$, a proposal move γ' is drawn from $T_*(\gamma'; \gamma^{[t]})$, a discrete uniform distribution over $\text{nb}d(\gamma^{[t]})$. The proposed move is accepted with probability

$$\alpha_* = \min \left\{ 1, \frac{p(y|\gamma')p(\gamma')}{p(y|\gamma^{[t]})p(\gamma^{[t]})} \right\},$$

which favors rejecting moves away from local modes even when the moves away would take the chain to more likely neighborhoods. In addition to these advantages, SSS outperforms MC^3 for datasets with large p due to the nature of their respective proposal distributions. Under the MC^3 proposal distribution, if the current model $\gamma^{[t]}$ is of dimension k , then the probability of proposing a model of dimension k is $1/(p+1)$, of proposing a model of dimension $k-1$ is $k/(p+1)$ and of proposing a model of dimension $k+1$ is $(p-k)/(p+1)$. When p is large relative to k , a model of dimension $k+1$ will be proposed most of the time, hampering the ability of MC^3 to move freely about the model space. In our experience with large datasets, even when using sparsity inducing priors such as described in Section 3.3, MC^3 algorithms tend to wander around high dimensional regions of the model space that have little posterior support. SSS avoids this problem by including γ° in $\text{nb}d(\gamma)$ and using the dimension balancing sampling step.

So far we have discussed MCMC algorithms based on a modified version of SSS. Directly converting SSS into a Metropolis-Hastings algorithm is complicated by the two stage sampling process used to balance dimension in the proposal distribution. The acceptance probability,

$$\alpha = \min \left\{ 1, \frac{p(\gamma'|y) T(\gamma^{[t]}; \gamma')}{p(\gamma|y) T(\gamma'; \gamma^{[t]})} \right\},$$

requires calculation of the transition probabilities, which in turn requires marginalizing over the two dimensions not sampled in the second stage. For example, if the sampled proposal γ' is from the addition set γ^+ , then the required forward transition probability is

$$T(\gamma'; \gamma^{[t]}) = \sum_{\gamma_*^o \in \gamma^o} \sum_{\gamma_*^- \in \gamma^-} \left[\frac{p(\gamma'|y)}{p(\gamma'|y) + p(\gamma_*^o|y) + p(\gamma_*^-|y)} \right] \times \left[\frac{p(\gamma'|y)}{\sum_{\mathbf{u} \in \gamma^+} p(\mathbf{u}|y)} \cdot \frac{p(\gamma_*^o|y)}{\sum_{\mathbf{v} \in \gamma^o} p(\mathbf{v}|y)} \cdot \frac{p(\gamma_*^-|y)}{\sum_{\mathbf{w} \in \gamma^-} p(\mathbf{w}|y)} \right], \quad (10)$$

where $p(\gamma|y)$ can be replaced by $S(\gamma)$ as the normalizing constants cancel. The large summation in (10) makes computation of both the forward and backward proposal probabilities undesirable. We do not view this as problematic, though, as computation of these probabilities is only needed to compute the acceptance probability, α . Because we are sampling proposals based on the restricted posterior distribution, and because we do not plan to use the resulting chain for Monte Carlo integration, it seems inefficient to reject a move. We prefer to treat SSS as a stochastic search tool that is similar to a Metropolis-Hastings algorithm and use SSS to thoroughly explore regions of high posterior probability.

4.5 Comparison to Gibbs Sampling

Gibbs sampling (Gelfand and Smith, 1990) is a particular MCMC algorithm that has been adapted for use in model space exploration problems. George and McCulloch (1997) and Smith and Kohn (1996, 1997) describe how to construct Gibbs samplers over a model space in the conjugate setting where $p(y|\gamma)$ is available in closed form. A one-at-a-time, fixed-

scan Gibbs sampler creates a sequence of models $\gamma^{[1]}, \gamma^{[2]}, \dots$, by updating the components of γ by sampling from $p(\gamma_j | \gamma_{-j}, y) \propto p(y | \gamma) p(\gamma_j | \gamma_{-j})$ for $j = 1, \dots, p$ at each iteration.

We implemented both SSS and the Gibbs sampler for the Keck dataset described in Section 4.3, using the observed rather than the simulated data. In both cases, the sparsity inducing prior $\pi = 10/p$ was used, and both search methods were run for 40,000 iterations. The Gibbs sampler evaluates p models per iteration, while SSS evaluates $p + k(p - k)$ models per iteration where k is the size of the current model at that iteration. As SSS spent most of its time around two and three variable models, this results in about $2p$ or $3p$ additional models evaluated per iteration for SSS compared to Gibbs; the resulting run time for Gibbs, however, was 16.5 hours, compared to just under 12 hours for SSS, and so comparing the two runs is fair from a compute time perspective. The top 1,000,000 models were saved from each run, along with the time at which they were first visited. Figure 4 shows the accumulated posterior mass in the set of 1,000,000 models for each run as a function of time, normalized by the total amount of mass in the set found by SSS. The amount of posterior mass in the set for Gibbs was 79.86% of the total mass in the set for SSS. We see that SSS moved to the high posterior density regions much faster than Gibbs, which took more time wandering around less interesting regions of the model space.

5 SSS Example Using Gene Expression Data

5.1 Data and Prediction Analysis Context

The data are coupled gene expression and lymph node positivity status in human breast cancers. From a data base of about 350 cases, we identify those that were clinically defined as low risk for disease recurrence, or death from disease, in terms of lymph node negativity (no evidence of cancer metastasis in the axillary lymph nodes) at the point of surgery; these patients are compared to those that are in a generally far higher risk group, i.e., those with

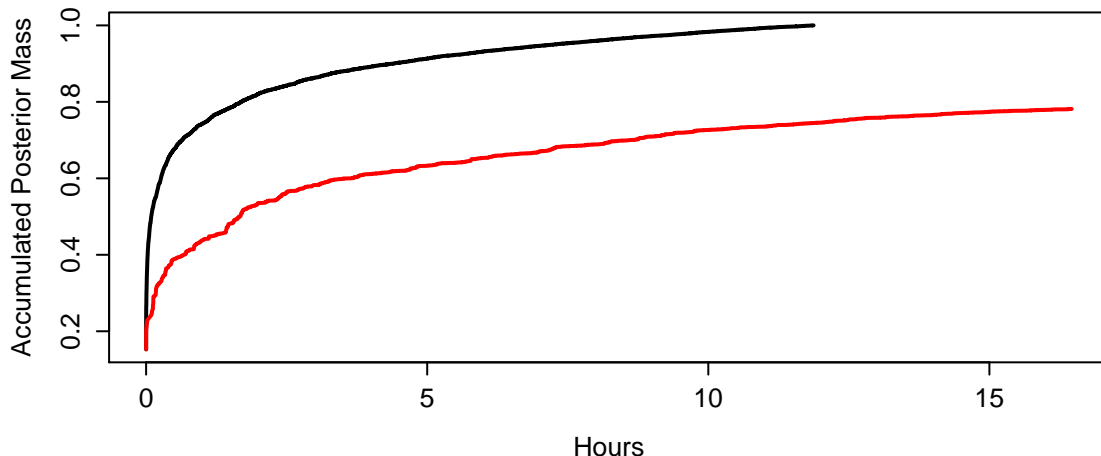


Figure 4: Accumulated posterior mass by time for SSS (black line) and Gibbs (red line).

at least nine nodes in the axillary regions showing evidence of cancer metastasis. This analysis follows previous work and relates to the general interest in the potential for tumor derived gene expression profiles to aid in prognosis – in this case, improved prediction of low versus high risk based on genomic information could feed into decisions about post-surgical treatments (West et al., 2001; Huang et al., 2002; Nevins et al., 2003; Huang et al., 2003; Pittman et al., 2004). Prediction of lymph node status based on gene expression profiles is a challenging problem, due to the complex heterogeneity of the disease in terms of genetic/genomic and environmental factors, and also as a result of the levels of experimental and technical noise in gene expression data. Advances in our ability to better predict the state would be of substantial interest in clinical cancer genomics.

The data consist of $n = 148$ samples with $n_0 = 100$ low risk (node negative) and $n_1 = 48$ high risk (high node positive cases). Gene expression data is available on Affymetrix HU95aV2 oligonucleotide microarrays, which were processed using the current standard RMA method (Irizarry et al., 2003a,b), to generate summary estimates of expression levels of each gene in each sample. This primary RMA data was then further screened and normalized, and we selected a total of 4,512 genes showing evidence of more than trivial variation

above the noise levels. In addition to these candidate predictors, each patient has a number of traditional clinical factors that are available, including estimate tumor size in centimeters and protein assay-based estrogen receptor (ER) status, coded as a binary covariate. Using the gene expression data together with these two clinical factors thus provides $p = 4,514$ candidate predictors.

5.2 Small Subsets Regression Analysis

We use binary regression models as described in Section 3.2 to both assess the relative importance of the individual genes and clinical factors in the context of lymph node invasion and to serve as a predictive model. We take $y_i = 0$ to denote node negative cases, and $y_i = 1$ to denote advanced nodal metastasis. The full vector of covariates, \mathbf{x}_i , is a $4,515 \times 1$ vector consisting of an intercept term, the two clinical variables (tumor size and ER status), and then the 4,512 gene expression variables.

As our focus is on sparse models, we take the prior distribution over the model space to be as described in Section 3.3 with $\pi = 10/4,514$. For $p(\beta|\gamma)$, we take $\tau = 1$ as described in Section 3.2. After running SSS for 20,000 iterations saving the top 100,000 models evaluated, we combined the results via the model averaging techniques in Section 3.

5.3 Results

The top 100,000 models evaluated contain a mix of one through seven variable models, as shown in Table 2. We compute a measure of posterior importance of model size, $|\gamma|$, in a similar fashion to the computation of the posterior importance of each individual variable based on (5):

$$\tilde{p}(|\gamma| = k) = C^{-1} \sum_{\gamma \in \Gamma^*} \mathbf{1}_{\{|\gamma|=k\}} p(y|\gamma) p(\gamma),$$

where $|\gamma|$ in the context of binary regression refers to the number of predictors in the model minus the intercept term. Under our model specification, the data give most support to small subset regressions of size five, six and four, in that order. No models of size eight or greater were found by SSS to belong in the list of top models.

Conditionally on Γ^* , eight genes were found to have posterior inclusion probability (5) greater than 0.10, as shown in the diagonal entries of Table 3. These genes dominate the list of models, as most of the four, five and six variable models include some subset of these genes. The most important gene, RGS3, occurs in almost all of the models. We have also computed pairwise importance measures according to

$$\tilde{p}(\gamma_i = \gamma_j = 1|y) = C^{-1} \sum_{\gamma \in \Gamma^*} \mathbf{1}_{\{\gamma_i = \gamma_j = 1\}} p(y|\gamma) p(\gamma).$$

These values are reported for the top eight variables in the off diagonal entries of Table 3, and confirm that models consisting of the top four variables dominate the list. Indeed, the four-way inclusion probability for the top four variables is 0.244, just less than a third of the total mass for five, six and seven variable models.

To assess the fit of the model, we computed model averaged mean probabilities p_i and associated 80% intervals using the top ten models. Figure 5 plots these model averaged fitted values vs. the linear predictor $\log(p_i/(1-p_i))$, which serves as a linear risk index. The fitted values have been corrected for the baseline incidence rate of 32.4%, making $p_i = 0.5$ the reference point. The model fit is quite good; 95.8% of the red points (true positives) are above 0.5, and 89% of the blue points (true negatives) are below 0.5. Further, 70.8% of the lower red quantiles are above 0.5, and 66.7% of the upper blue quantiles are below 0.5. These are patients for whom we have fit properly with high probability.

To assess the way in which the top genes combine across models in a predictive context, we took the genes that comprise the top ten models (a total of 18 genes) and created two “metagenes”, the first two principal components from a singular value decomposition of the 18 genes. Figure 6 shows the association between these two metagenes and the model

k :	1	2	3	4	5	6	7
# of models	1	54	1,311	11,838	54,597	30,619	1,580
$\tilde{p}(\gamma = k y)$	< 0.001	0.001	0.020	0.184	0.534	0.253	0.007

Table 2: Posterior probability of model size, k , conditioned on the top 100,000 models discovered by SSS.

	RGS3	DXYS155E	ATP6V1F	MGC8721	VDAC1	GEM	WSB1	PRRG1
RGS3	0.991	0.716	0.495	0.351	0.169	0.133	0.125	0.110
DXYS155E		0.716	0.454	0.319	0.069	0.069	0.121	0.101
ATP6V1F			0.498	0.250	0.010	0.045	0.108	0.039
MGC8721				0.352	0.016	0.042	0.054	0.006
VDAC1					0.171	0.037	0.001	0.047
GEM						0.134	*	*
WSB1							0.125	0.002
PRRG1								0.110

Table 3: Genewise and pairwise inclusion probabilities. The diagonal elements in boxes are the quantities $\tilde{p}(\gamma_j = 1|y)$, and the off-diagonal elements are the quantities $\tilde{p}(\gamma_i = \gamma_j = 1|y)$. The character “*” indicates a value < 0.001.

averaged linear predictor computed above. We expect a concordance between this empirical metagene and the averaged predictions, but it is evident from the variation in the scatter plot that the complex, data-weighted mixing over the set of regression models is generating predictions that are not simply captured by a single linear fit – the metagene – to the selected set of most interesting predictor variables.

To assess aspects of the predictive fit of the overall model, we performed a leave-one-out cross validation prediction analysis. Leaving out observation i , we recompute model probabilities and derive model averaged predictions of the response probability for case i based on the remaining observations. A histogram of predicted risk index $\log(p_i/(1 - p_i))$ is shown in Figure 7. On the basis of simple thresholding of these point estimates at zero, corresponding to a simple thresholding of the corresponding point predictions of metastasis, the analysis indicates an approximate sensitivity of 79.2% and a specificity of 76%. This level of predictive discrimination is quite high and suggests promise for the approach relative to prior analyses on much smaller and selected subsets of patients (West et al., 2001; Huang et al., 2003).

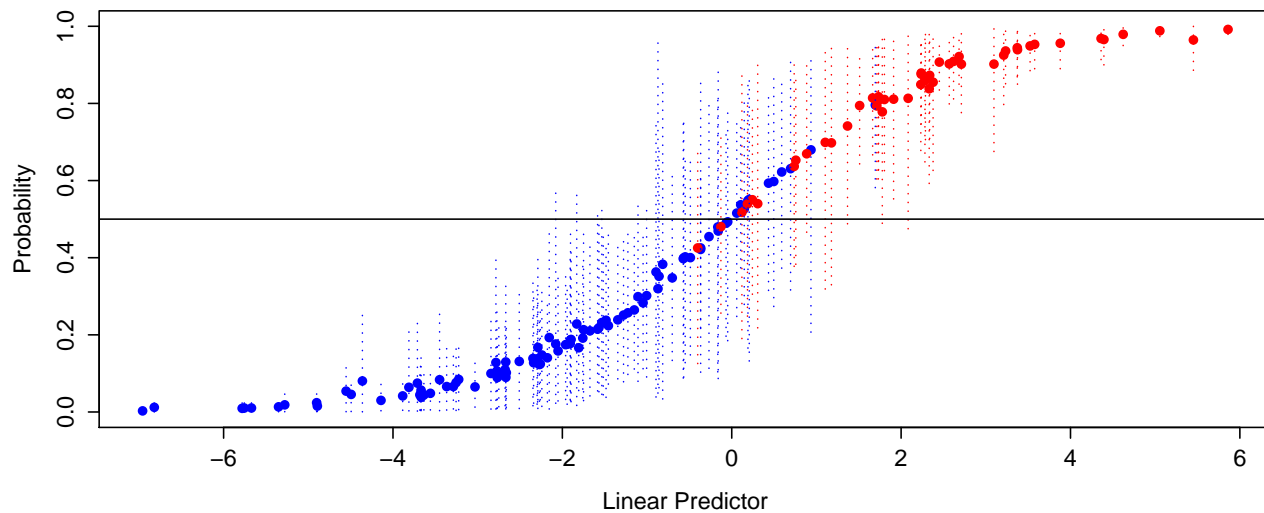


Figure 5: Model averaged fitted values based on the top ten models, with associated 80% intervals. The red points indicate $y_i = 1$ and the blue indicate $y_i = 0$.

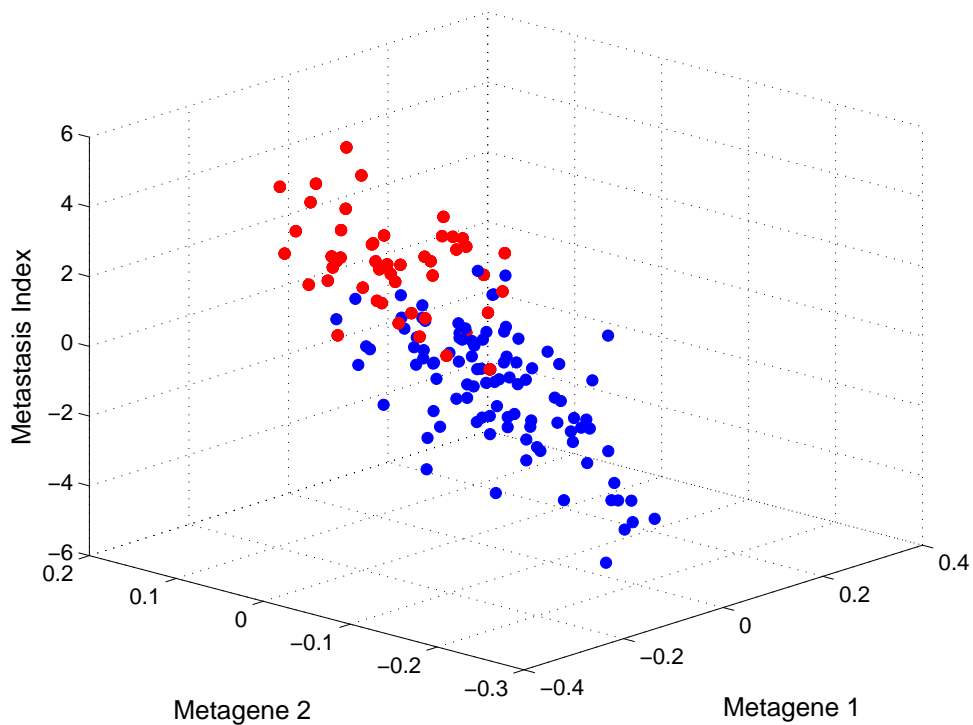


Figure 6: Association of metagenes with the model averaged metastasis index (linear predictor) based on the eighteen genes comprising the top ten models. Red points indicate $y_i = 1$, blue points indicate $y_i = 0$.

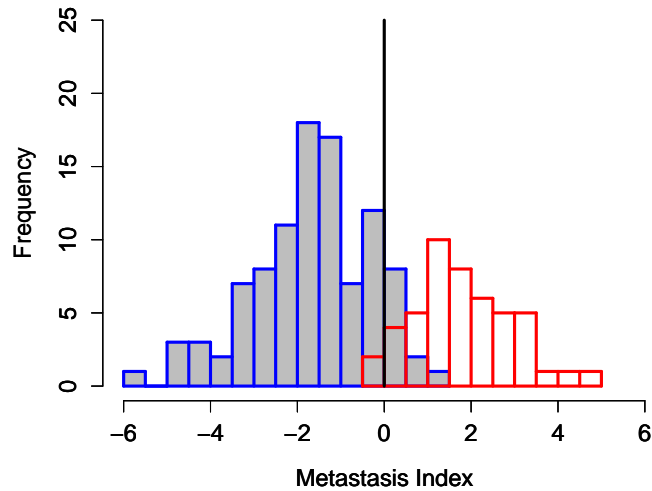


Figure 7: Histograms of the leave one out cross validated predictions on the linear predictor scale. The blue, shaded histogram is for the true negatives and the red histogram is for the true positives.

6 Additional Comments

We have presented a novel stochastic search approach for exploring regression model spaces using the power of distributed computing to allow consideration of potentially tens of thousands of possible predictor variables. The shotgun stochastic search approach is quite general and, in addition to the linear and binary regression models that we have considered here, can be applied to any generalized linear regression model so long as the marginal likelihood can be evaluated or approximated. Some current and recent analyses are demonstrating the ability of SSS to rapidly identify multiple regions of model space exhibiting high posterior probability – or, more generally, high model “scores” – and the utility of the approach in contexts where traditional search and MCMC methods are simply ineffective due to both dimension and the subtlety of predictive relationships in the context of noise and complex patterns of collinearity. One recent example using linear regression in a cancer survival genomics study of Rich et al. (2005) has illustrated this in connection with both predictive utility and variable selection/identification in a challenging context.

We note that applications outside of regression are possible as well, e.g. Jones et al. (2005) consider SSS in the context of Gaussian graphical model determination, and we anticipate further developments in that area as well as others. Some of the currently topical questions of interest include improved computational implementations and also study of more general classes of prior distributions over model spaces.

Software implementing the SSS analysis for linear regression, binary regression using logistic models, and also survival regression using Weibull models, is available for use by interested readers. The code is written in C++ and utilizes MPI for implementation in a distributed Beowulf cluster environment. The code may be modified to implement other sampling distributions, and also to run in serial implementation. Full details are available at www.isds.duke.edu/research/software under the SSS item listing.

Acknowledgements

We acknowledge support of the National Science Foundation (grants DMS-0102227 and DMS-0342172), the National Institutes of Health (grants NHLBI 1P01-HL-73042-02), and the W.M. Keck Foundation. Any opinions, findings and conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of the NSF, NIH or Keck Foundation. The authors are also grateful to Quanli Wang of Duke University for advice and assistance on computational matters, and to the editor for his comments on the original version of the paper.

A Transition Probabilities for Orthogonal Designs

The transition probabilities for the Markov chain $\{Z_t\}$ described in Sections 4.2 for an orthogonal design are

$$\Pr(Z_{t+1} = j | Z_t = i) = \begin{cases} (11) & \text{if } j = i - 1, 0 < i < k, \\ (12) & \text{if } j = i \neq k, \\ (13) & \text{if } j = i + 1, i < k, \\ 1 & \text{if } j = i = k, \\ 0 & \text{o.w.,} \end{cases}$$

where the referenced equations are

$$\frac{1}{1 + \left(\frac{k-i}{i}\right) \left(\frac{p-2k+2i}{p-2k+i}\right) \left(1 - \frac{\epsilon^2 - \lambda^2}{b_i}\right)^\nu + \left(\frac{(k-i)^2}{i(p-2k+i)}\right) \left(\frac{b_i - (\epsilon^2 - \lambda^2)}{b_i + (\epsilon^2 - \lambda^2)}\right)^\nu}, \quad (11)$$

$$\frac{1}{1 + \left(\frac{i}{k-i}\right) \left(\frac{p-2k+i}{p-2k+2i}\right) \left(1 - \frac{\epsilon^2 - \lambda^2}{b_i}\right)^{-\nu} + \left(\frac{k-i}{p-2k+2i}\right) \left(1 + \frac{\epsilon^2 - \lambda^2}{b_i}\right)^{-\nu}}, \quad (12)$$

$$\frac{1}{1 + \left(\frac{i(p-2k+i)}{(k-i)^2}\right) \left(\frac{b_i + (\epsilon^2 - \lambda^2)}{b_i - (\epsilon^2 - \lambda^2)}\right)^\nu + \left(\frac{p-2k+2i}{k-i}\right) \left(1 + \frac{\epsilon^2 - \lambda^2}{b_i}\right)^\nu}, \quad (13)$$

where $b_i = a - (k - i)\epsilon^2 - i\lambda^2$.

References

- Brown, P. J., Vannucci, M., and Fearn, T. (1998a). “Bayesian wavelength selection in multicomponent analysis.” *Journal of Chemometrics*, 12, 173–182.
- (1998b). “Multivariate Bayesian variable selection and prediction.” *Journal of the Royal Statistical Society, Series B*, 60, 672–641.
- Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). “Bayesian CART Model Search.” *Journal of the American Statistical Association*, 93, 935–948.

- Clyde, M. and George, E. I. (2004). “Model Uncertainty.” *Statistical Science*, 19, 81–94.
- DiCiccio, T. J., Kass, R. E., and Wasserman, L. (1997). “Computing Bayes factors by combining simulation and asymptotic approximations.” *Journal of the American Statistical Association*, 92, 903–915.
- Dobra, A., Hans, C., Jones, B., Nevins, J., Yao, G., and West, M. (2004). “Sparse graphical models for exploring gene expression data.” *Journal of Multivariate Analysis*, 90, 196–212.
- Furnival, G. M. and Wilson, R. W. (1974). “Regression by leaps and bounds.” *Technometrics*, 16, 499–511.
- Gelfand, A. E. and Smith, A. F. M. (1990). “Sampling-based approaches for calculating marginal densities.” *Journal of the American Statistical Association*, 85, 398 – 409.
- George, E. I. and McCulloch, R. E. (1993). “Variable selection via Gibbs sampling.” *Journal of the American Statistical Association*, 88, 881–889.
- (1997). “Approaches for Bayesian variable selection.” *Statistica Sinica*, 7, 339–373.
- Geweke, J. (1996). “Variable selection and model comparison in regression.” In *Bayesian Statistics 5* Editors: J.M. Bernardo, J.O. Berger, A.P. Dawid and A.F.M. Smith, 609–620. Oxford Press.
- Green, P. J. (1995). “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination.” *Biometrika*, 82, 711–732.
- Huang, E., Chen, S., Dressman, H., Pittman, J., Tsou, M. H., Horng, C. F., Bild, A., Iversen, E. S., Liao, M., Chen, C. M., West, M., Nevins, J. R., and Huang, A. T. (2003). “Gene expression predictors of breast cancer outcomes.” *The Lancet*, 361, 1590–1596.
- Huang, E., West, M., and Nevins, J. R. (2002). “Gene expression profiles and predicting clinical characteristics of breast cancer.” *Hormone Research*, 58, 55–73.

- Irizarry, R. A., Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B., and Speed, T. P. (2003a). “Summaries of Affymetrix GeneChip probe level data.” *Nucleic Acids Research*, 31, e15.
- Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003b). “Exploration, normalization, and summaries of high density oligonucleotide array probe level data.” *Biostatistics*, 2, 249–264.
- Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, C., and West, M. (2005). “Experiments in stochastic computation for high-dimensional graphical models.” *Statistical Science*, 20, 388–400.
- Madigan, D. and York, J. (1995). “Bayesian Graphical Models for Discrete Data.” *International Statistical Review*, 63, 215–232.
- Nevins, J. R., Huang, E. S., Dressman, H., Pittman, J., Huang, A. T., and West, M. (2003). “Towards integrated clinico-genomic models for personalized medicine: Combining gene expression signatures and clinical factors in breast cancer outcomes prediction.” *Human Molecular Genetics*, 12, 153–157.
- Pittman, J., Huang, E., Dressman, H., Horng, C. F., Cheng, S. H., Tsou, M. H., Chen, C. M., Bild, A., Iversen, E. S., Huang, A. T., Nevins, J. R., and West, M. (2004). “Integrated modeling of clinical and gene expression information for personalized prediction of disease outcomes.” *Proceedings of the National Academy of Sciences*, 101, 8431–8436.
- Raftery, A. E. (1995). “Bayesian model selection in social research.” *Sociological Methodology*, 25, 111–163.
- Raftery, A. E., Madigan, D., and Hoeting, J. (1997). “Bayesian Model Averaging for Linear Regression Models.” *Journal of the American Statistical Association*, 92, 1197–1208.
- Rich, J. N., Hans, C., Jones, B., Iversen, E. S., McClendon, R. E., Rasheed, B. K. A., Dobra,

- A., Dressman, H. K., Bigner, D. D., Nevins, J. R., and West, M. (2005). “Gene expression profiling and genetic markers in glioblastoma survival.” *Cancer Research*, 65, 4051–4058.
- Smith, M. and Kohn, R. (1996). “Nonparametric regression using Bayesian variable selection.” *Journal of Econometrics*, 75, 317–343.
- (1997). “A Bayesian approach to nonparametric bivariate regression.” *Journal of the American Statistical Association*, 92, 1522–1535.
- Tadesse, M. G., Sha, N., and Vannucci, M. (2005). “Bayesian variable selection in clustering high-dimensional data.” *Journal of the American Statistical Association*, 100, 602–617.
- West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Marks, J. R., and Nevins, J. R. (2001). “Predicting the clinical status of human breast cancer utilizing gene expression profiles.” *Proceedings of the National Academy of Sciences*, 98, 11462–11467.
- Wong, F., Carter, C. K., and Kohn, R. (2003). “Efficient estimation of covariance selection models.” *Biometrika*, 90, 809–830.