Shotgun Stochastic Search for “Large p” Regression

Original by Hans, C., Dobra, A., & West, M.

Review by Nayeon Kwon & Yejin Jeong

December 8, 2021

Overview

1

Introduction

1 Introduction

Motivation

- Model selection raises modeling and computational challenges as the number of candidate predictor variables increases.
- Standard approaches(e.g. stepwise methods, best subset regression, MCMC) methods are often infeasible or ineffective when predictor variables increases.
- This paper describes a shotgun stochastic search(SSS) approach that explores "interesting" regions of the resulting high-dimensional model spaces and quickly identifies regions of high posterior probability over models.

1 Introduction

Compare MH algorithm with SSS

MH algorithm

- Traditional MCMC methods move sequentially from one model to a new model and so do not exploit the opportunity to effectively explore the model space in neighborhood of known "good" models.
- MCMC methods move only toward individual models of higher probability.

Shotgun Stochastic Search

- SSS explores the vast discrete space of regression models by evaluating and recording many candidate models in parallel at each iteration.
- SSS is designed to move toward and aggressively explore regions of regression model space that contain multiple higher-probability models. It automatically seeks out many good models near good models.

2

Regression Model Shotgun Stochastic Search

2.1 What is Shotgun Stochastic Search?

RMSSS

- Regression Model Shotgun Stochastic Search
- A regression model specific implementation of general class of SSS methods

SSS

- Shotgun Stochastic Search
- Main component : neighborhood & the model move (sampling) strategy

✓ Iterative
✓ local-move
✓ neighborhood

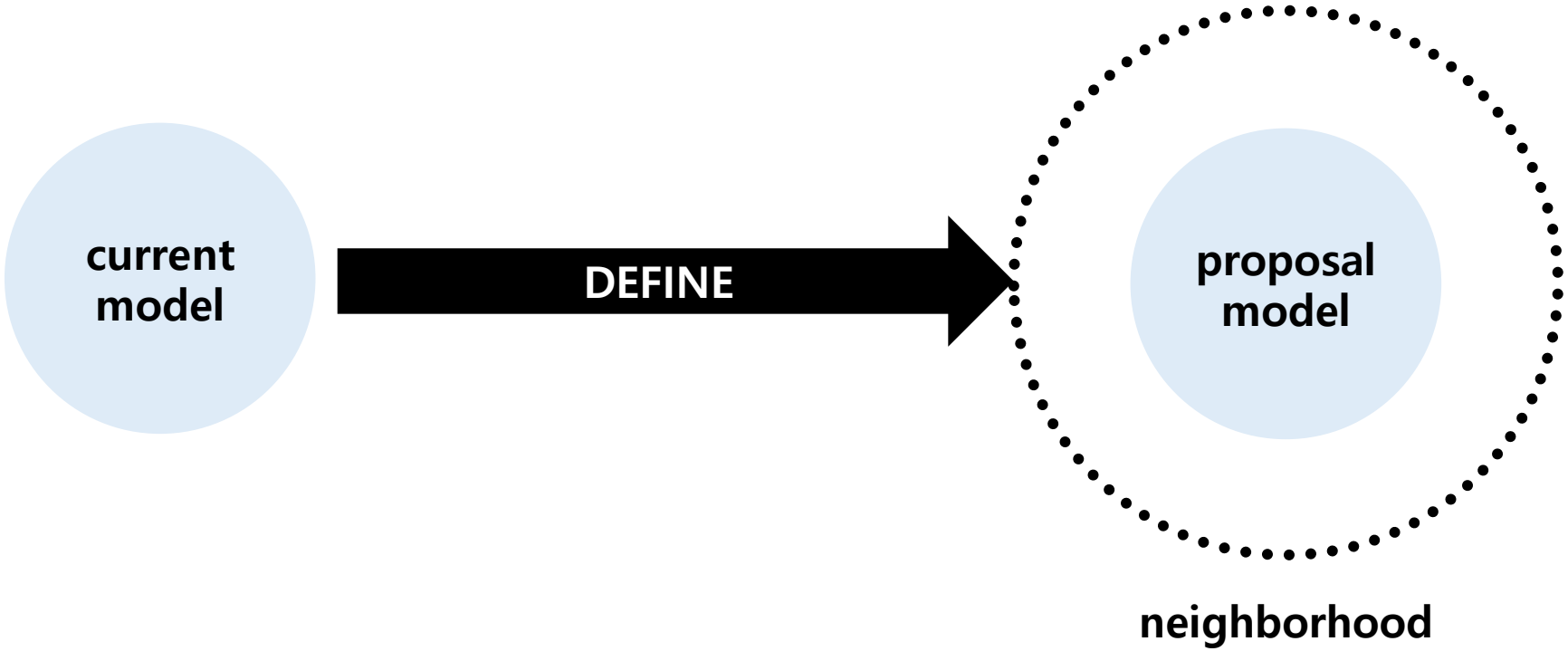
} 3 steps

2.1 What is Shotgun Stochastic Search?

Step 1 Use the current model to define a neighborhood of proposal models.

Step 2 Evaluate each proposal model in this neighborhood in parallel.

Step 3 Choose a new current model from the proposals.

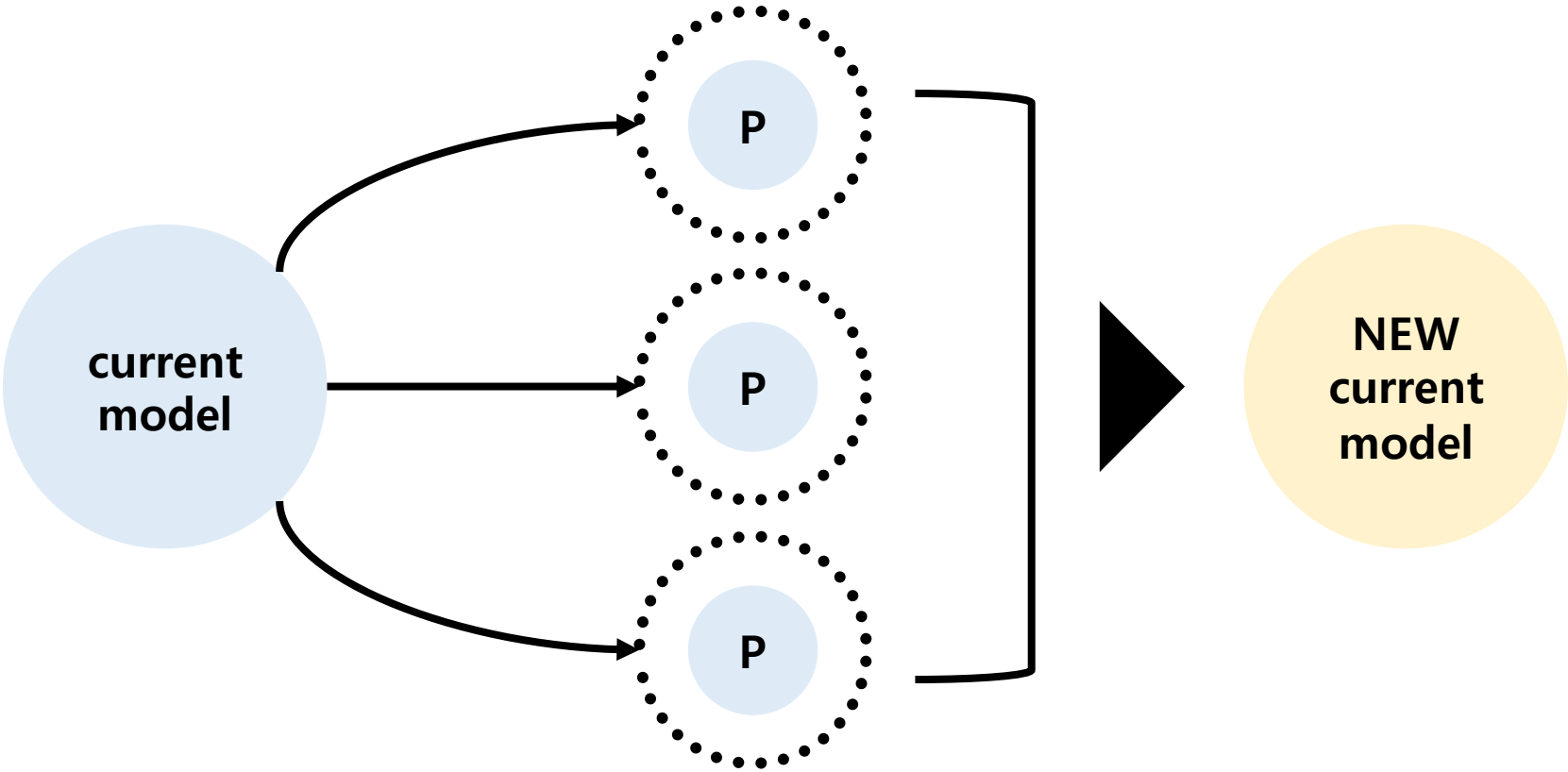


2.1 What is Shotgun Stochastic Search?

Step 1 Use the current model to define a neighborhood of proposal models.

Step 2 Evaluate each proposal model in this neighborhood in parallel.

Step 3 Choose a new current model from the proposals.



2.1 What is Shotgun Stochastic Search?

The Benefit of SSS

1

Every candidate variable is evaluated in the context of many different regression models.

2

Each proposal models can be evaluated independently on separate processors.

2.2 RMSSS

Setting

- p : the total number of possible predictor variables
- γ : $p \times 1$ indicator vector with $\gamma_j = \begin{cases} 1 \ \ if \ \ j \ \in regression\ model \\ 0 \ \ if \ \ j \ \notin regression\ model \end{cases}$
- k : dimension of current model γ

$$\boldsymbol{nbd}(\boldsymbol{\gamma}) = \{\boldsymbol{\gamma}^{+}, \boldsymbol{\gamma}^{\circ}, \boldsymbol{\gamma}^{-}\}$$

- ✓ **addition moves($\boldsymbol{\gamma}^{+}$)** : models obtained by adding any one of the $p-k$ remaining predictor variables
- ✓ **replacement moves($\boldsymbol{\gamma}^{\circ}$)** : models obtained by replacing any current variable with any one of $p-k$ remaining predictor variables
- ✓ **deletion moves($\boldsymbol{\gamma}^{-}$)** : models obtained by deleting any one current variable

2.2 RMSSS

Example

- The total number of possible predictor variables $p = 5$
- Current model is $\{x_1, x_3, x_4\}$

$$\gamma^{+} = \bigcup_{j \in \{2,5\}} \{x_1, x_3, x_4, x_j\}$$

$$\gamma^{\circ} = \bigcup_{j \in \{2,5\}} \{\{x_3, x_4, x_j\}, \{x_1, x_4, x_j\}, \{x_1, x_3, x_j\}\}$$

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

2.2 RMSSS

The Number of Samples

- When $2 \leq k \leq p$,

$$|\gamma^+| = p - k$$

$$|\gamma^\circ| = k(p-k)$$

$$|\gamma^-| = k$$

- If p is large,

$$k \ll p - k \ll k(p-k)$$

$$\Leftrightarrow \quad |\gamma^-| \ll |\gamma^+| \ll |\gamma^\circ| \quad \text{(Problematic for sampling)}$$

2.2 RMSSS

Sampling

- To move across dimension effectively, let's break sampling into 2 steps.

Step 1 Sample other three models γ_*^+, γ_*° , γ_*^- from each γ^+, γ°, γ^-

Step 2 One of three is selected

2.2 RMSSS

Regression Model Shotgun Stochastic Search Schema

- Let γ : regression model, $S(\gamma)$: corresponding score of γ
- Given γ^0 : starting model, $\Gamma^* = \{\gamma^{[0]}\}$, B : max number of elements of Γ^*
- Iterate in $t = 1, \ldots, T$ with following steps

Step 1 Compute $S(\gamma)$ for all $\gamma \in nbd(\gamma^{[t]})$, constructing γ^+, γ°, γ^-

Update Γ^* to be $\Gamma^* \cup nbd(\gamma^{[t]})$

If $|\Gamma^*| > B$, then remove $|\Gamma^*| - B$ lowest scoring models

Step 2 Sample other three models γ_*^+, γ_*° , γ_*^- from each γ^+, γ°, γ^- with $S(\gamma)$ normalized with set.

Step 3 Sample $\gamma^{[t+1]}$ from $\{\gamma_*^+, \gamma_*^\circ, \gamma_*^-\}$ with $S(\gamma)$ normalized with set.

3

Comparison with Markov Chain Monte Carlo Methods

3.1 Relationship to Markov Chain Monte Carlo

The Aim of This Section

- In cases of high-dimensional parameter spaces, MCMC approaches are often used to identify the "best" models.
- In this section, we show that small changes to SSS result in an MCMC algorithm of a fundamentally different from than common MCMC approaches.

3.1 Relationship to Markov Chain Monte Carlo

Metropolis-Hastings Algorithm to Sample from a Discrete Distribution $P(x)$

$$T(x'; X_t) = \frac{p(x')I\left(x' \in N(x_t)\right)}{\sum_{s \in N(x_t)} P(s)} = \frac{Q(x')I\left(x' \in N(x_t)\right)}{\Sigma_{s \in N(x_t)} Q(s)}$$

- $P(x)$: a discrete distribution, where we can evaluate $P(x)$ up to a normalizing constant, $P(x) \propto Q(x)$.
- $T(x'; x_t)$: proposal distributions that sample from $P(x)$ restricted to a neighborhood $N(\cdot)$

The Acceptance Probability at Each Iteration

$$\alpha = \min\left\{1, \frac{\sum_{s \in N(x_t)} Q(s)}{\sum_{s \in N(x')} Q(s)}\right\}$$

- We can adapt the SSS algorithm to become a Metropolis-Hastings algorithm using the proposal distribution.

3.1 Relationship to Markov Chain Monte Carlo

Metropolized SSS

- $p(x_t) = p(\gamma^{[t]}|y)$
- $Q(x_t) = S(\gamma^{[t]}) = P(y|\gamma^{[t]})P(\gamma^{[t]})$
- $N(x_t) = nbd(\gamma^{[t]})$

Step 1-1 Perform step 1 at iteration t in SSS.

Step 1-2 Sample a proposal γ' from the discrete distribution $S(\cdot)$ normalized within $nbd(\gamma^{[t]})$.

Step 1-3

$$\gamma^{[t+1]} = \begin{cases} \gamma' & \text{with probability} \quad \alpha, \\ \gamma^{[t]} & \text{with probability} \quad 1-\alpha. \end{cases}$$

3.1 Relationship to Markov Chain Monte Carlo

MC^3 Algorithm (Madigan and York. 1995)

- MC^3 constructs a Markov chain over the model space by defining a neighborhood $nbd_*(\gamma^{[t]})=\gamma^+ \cup \gamma^- \cup \gamma^{[t]}$
- A proposal distribution T_* is defined by setting

$$T_*(\gamma';\gamma^{[t]}) = \begin{cases} 0 & for\ all\ \gamma' \notin\ nbd_*(\gamma^{[t]}) \\ constant & for\ all\ \gamma' \epsilon\ nbd(\gamma^{[t]}) \end{cases}$$

- As the MC^3 algorithm proceeds, if the chain is in state $\gamma^{[t]}$, then a proposed move γ' is drawn from $T_*(\gamma';\gamma^{[t]})$, a discrete uniform distribution over $nbd(\gamma^{[t]})$.
- The proposed move is accepted with probability

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

3.1 Relationship to Markov Chain Monte Carlo

Compare Metropolized SSS with MC^3 Algorithm

MC^3 Algorithm

- The acceptance probability α_* depends only on $\gamma^{[t]}$ and γ' and favors rejecting moves to lower-probability models

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

Metropolized SSS

- The acceptance probability α for Metropolized SSS, depends on the amount of posterior mass in the *neighborhoods* around $\gamma^{[t]}$ and γ', and favors moves to models away from local models, discouraging entrapment in particular regions of model space.

$$\alpha = \min\left\{1, \frac{\sum_{s\epsilon N(x_t)} Q(s)}{\sum_{s\epsilon N(x')} Q(s)}\right\}$$

3.2 Comparison with Markov Chain Monte Carlo Methods

SSS and Two MCMC Approaches

- Two MCMC approaches for model space exploration: MC^3 and Gibbs sampling
- This paper implemented SSS and both MC^3 and the Gibbs sampler for the Keck dataset (observed data)
- In both cases, we used the sparsity-inducing prior $\pi = 10/p$.
- The three runs represent approximately the same number of model evaluations.

SSS	MC^3	Gibbs sampler
• SSS was run for 40,000 iterations, which resulted in a total of 1,137,195,208 total model evaluations	• MC^3 evaluates one model per iteration for 1,137,195,208 iterations.	• Gibbs sampler evaluates $p = 8{,}408$ models per iteration, for 135,252 iterations.

Table: The number of iteration for each methods

3.2 Comparison with Markov Chain Monte Carlo Methods

Model Evaluations for SSS and Two MCMC Methods

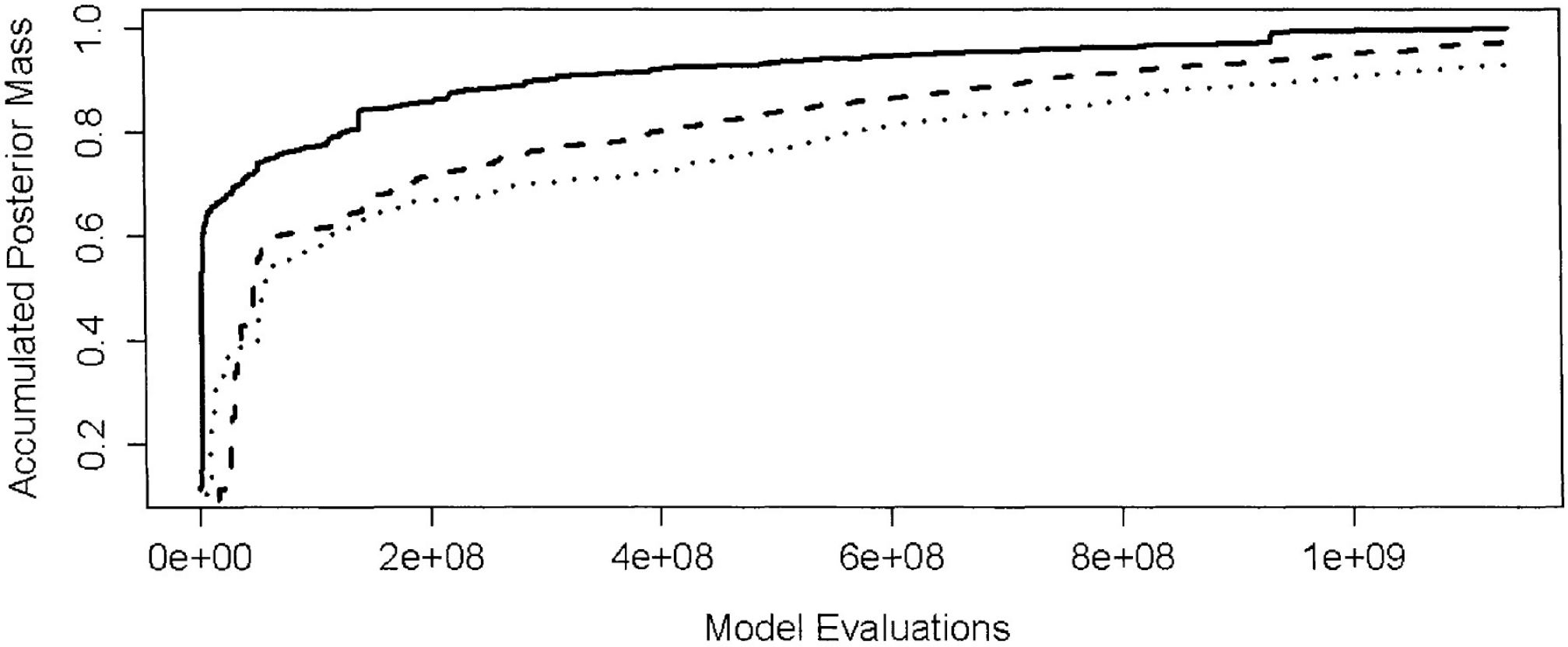


Figure 1. Accumulated Posterior Mass for SSS (——), Gibbs (- - - - -), and MC^3 (· · · · · ·).

4

Simulation Study

4.1 The Null Model Scenario

The Null Model Scenario

- It examine how SSS performs when the generating model is the null normal linear model.

Simulation Setting

- $n = 100$ from independent standard normal distributions
- $P = 5{,}000$ covariates from independent standard normal distributions
- The data were mean centered and standardized.

4.1 The Null Model Scenario

Model-Averaged Fitted Values

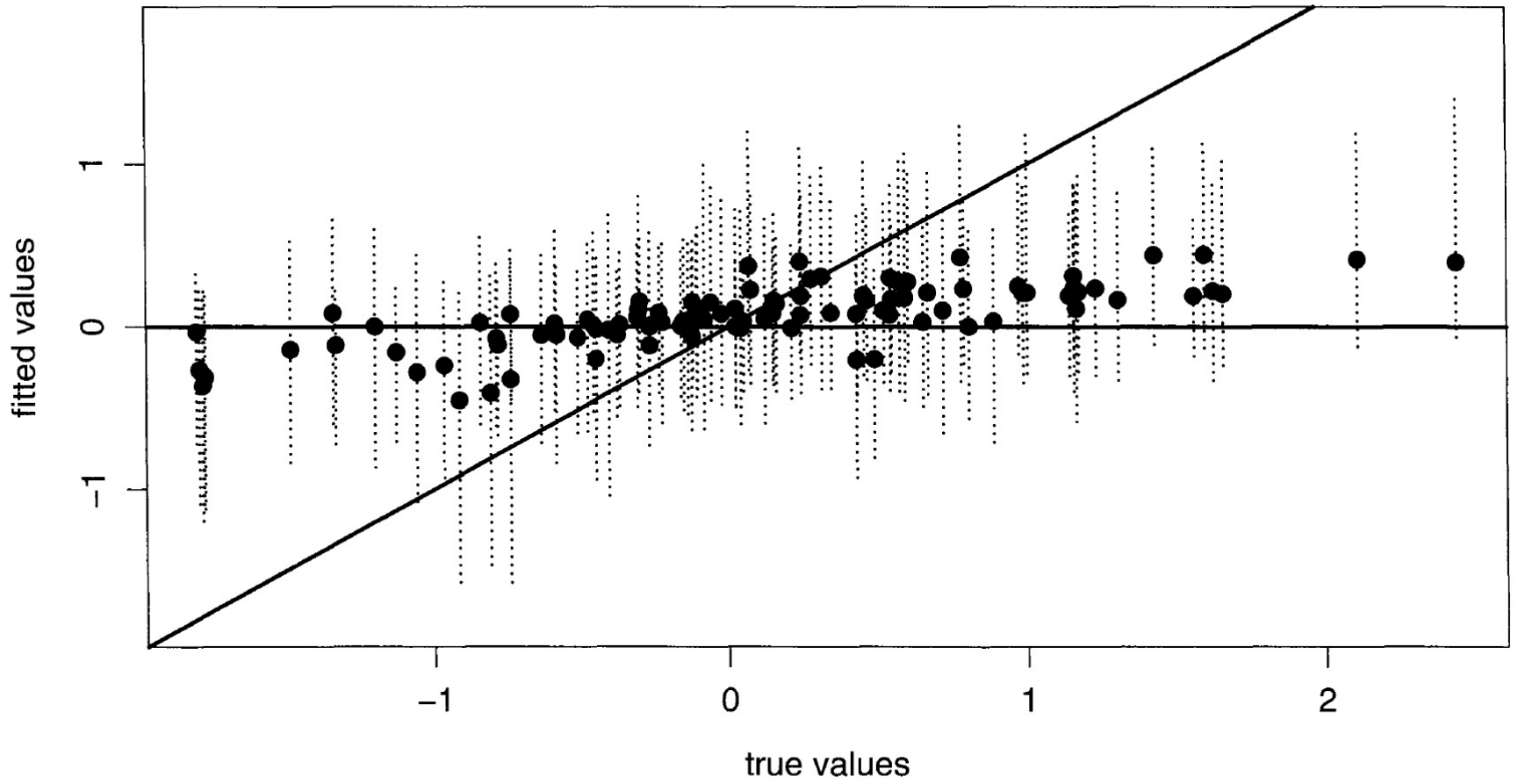


Figure 2. Model-Averaged Fitted Values When the True Model Is the Null Model, With 95% Intervals.

4.2 A Simulation Study Based on a Real Dataset

Simulation Setting

- The study group consists of 41 patients, each of which has gene expression data consisting of 8,408 genes from a tumor specimen.
- We selected four genes from the dataset as the variables composing the "true" model γ^*
  - A gene from RAS oncogene family, a glioblastoma-amplified sequence, a leukocyte protease inhibitor, and a CAT56 protein.
- 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

$$y_i^{(m)} = 1.3x_{i1} + .3x_{i2} - 1.2x_{i3} - .5x_{i4} + \varepsilon_i^{(m)}, \text{ for } i = 1, \dots, 41,$$

  where $\varepsilon_i^{(m)}$'s are iid mean-0 normal random variable with variance .5.
- The simulated outcomes were standardized to have mean 0 and unit variance within each of the 50 simulations.

4.2 A Simulation Study Based on a Real Dataset

Simulation Setting

- We ran SSS for the 50 simulated responses using datasets with increasing values of p.
- The datasets were constructed by first reordering the observed 41 x 8,408 data matrix X so that the four variables used in the simulation were designated variables 1, 2, 3, and 4.
- Prior distributions over the parameter space 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.
  - We assume priors on $\theta = (\beta, \sigma^2)'$
  - $p(\beta_\gamma | \sigma^2, \gamma) = N(0, \tau^{-1}\sigma^2 I_k)$ and $p(\sigma^2 | \gamma) = IG((\delta + k)/2, \tau/2)$, where β_γ is the vector of regression coefficients under a model γ with k variables.
- For the model space prior, we set $\pi = 4/p$ as in Section 3.3 to maintain focus on sparse models as p increases.

4.2 A Simulation Study Based on a Real Dataset

Simulation Results

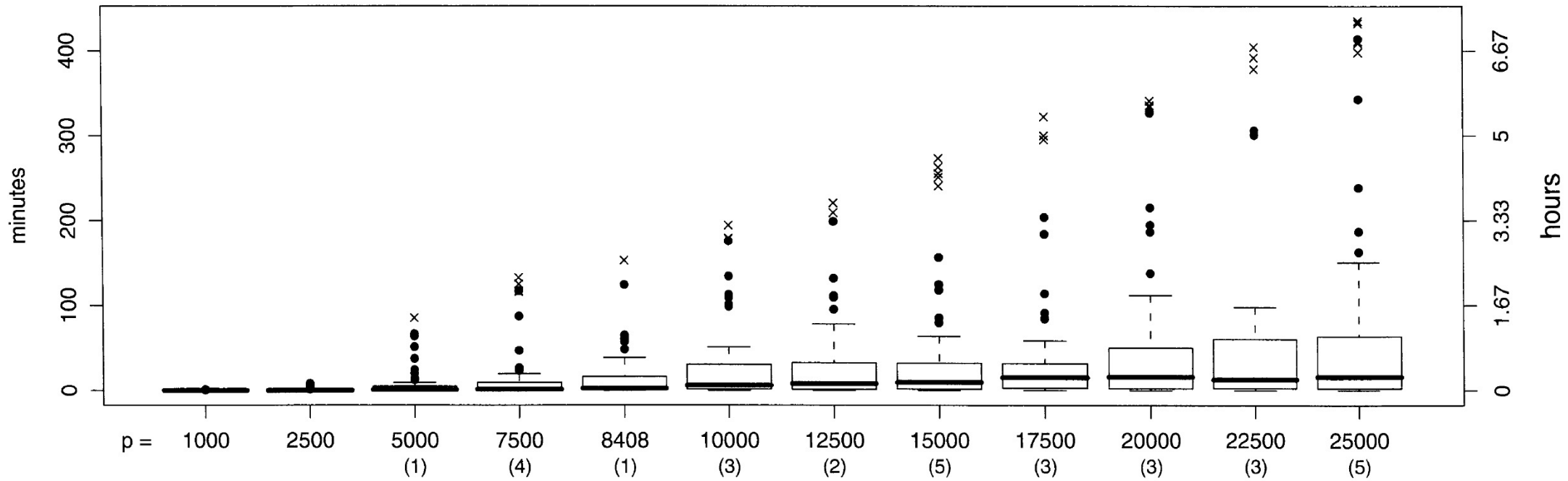


Figure 3. Time Required to Find the True Model for the Simulation Study Described in Section 4.4. The numbers in parentheses indicate the number of models not found by SSS in 10,000 iterations for a given dataset size p, denoted by × in the plot. The boxplots are based only on runs for which SSS found the true model.

5

Conclusion

5 Conclusion

Conclusion

Pros

- SSS explores the space of potentially very many models.
- There are various choices for selecting models and each process is independent.
- Metropolized SSS discourages entrapment in particular regions of model space.

Cons

- Current model should be existing for updating new model.
- It is necessary to sample from the posterior distribution when selecting models with SSS
- SSS does not consider the possibility that we can't find the true model when model space is large.

Reference

Hans, C., Dobra, A., & West, M. (2007). Shotgun Stochastic Search for “Large p” Regression. Journal of the American Statistical Association, 102(478), 507–516.