

Variable selection with VsusP

Nilson Chapagain, Debdeep Pati

1 Model and selection rule

For a Gaussian linear model, let Y be the response vector, X the design matrix and β the coefficient vector:

$$Y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I_n).$$

Here n is the number of observations and σ^2 is the residual variance. Shrinkage priors draw coefficients toward zero without setting them exactly to zero. VsusP uses the sequential two-means (S2M) method of Li and Pati (2017, Section 2.2.1) to select a subset from these posterior draws.

For each draw, S2M clusters absolute coefficients into two groups and saves the lower-mean group as the noise candidate set D . While the center gap exceeds b , it saves the current lower-mean group and splits that group again. When the gap is at most b , it retains the saved D , giving signal count $H = p - |D|$, where p is the number of coefficients. VsusP also stops when the candidate has at most two elements or identical magnitudes. It uses the modal count across draws, then selects the H largest posterior median absolute coefficients. Equal-frequency modes favor the smaller count; equal medians favor larger column indices.

2 Installation

```
install.packages("VsusP")
library(VsusP)
```

For the development version:

```
# Install devtools first if it is not available.
devtools::install_github("nilson01/VsusP-variable-selection-using-shrinkage-priors")
```

3 Selection from posterior draws

The rows of `Beta` are posterior draws; columns correspond to coefficients. Exclude any intercept column. These fixed draws show the selection steps without fitting a regression model.

```
Beta <- rbind(c(.01, .02, .03, .3, .31, 10, 11),
             c(-.01, -.02, -.03, -.3, -.31, -10, -11))
colnames(Beta) <- paste0("x", seq_len(ncol(Beta)))
set.seed(1)
result <- Sequential2MeansBeta(Beta, lower=.1, upper=.5, l=2)
data.frame(b=result$b.i, H=result$H.b.i)
#>      b H
#> 1 0.1 4
#> 2 0.5 2
```

`OptimalHbi()` displays this path. It does not select a threshold or compute an optimal value. In this illustration, we use $b = 0.5$ and its corresponding count.

```
OptimalHbi(result$b.i, result$H.b.i)
```

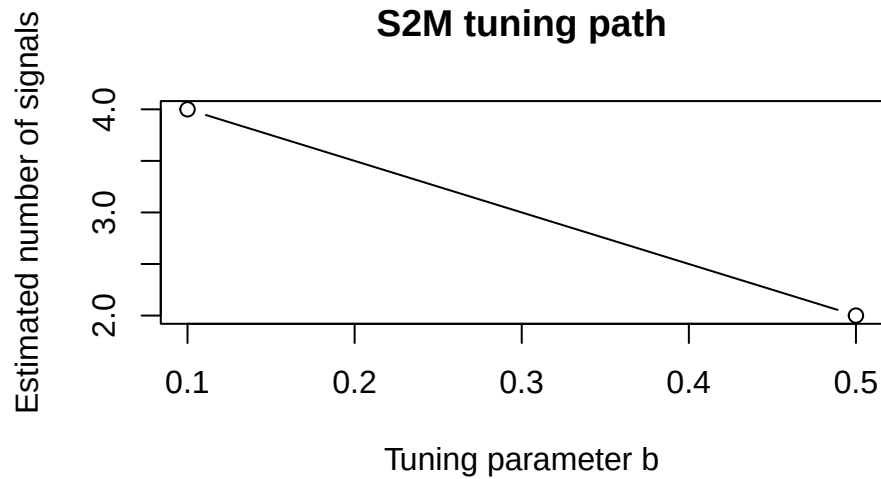


Figure 1: S2M signal counts for the fixed coefficient draws.

```
H <- result$H.b.i[2]
indices <- S2MVarSelection(Beta, H)
colnames(Beta)[indices]
#> [1] "x7" "x6"
S2MVarSelectionV1(result, H)
#> [1] 7 6
```

Both selection functions return column indices. `S2MVarSelectionV1()` uses the stored medians rather than recomputing them. `H=0` returns `integer(0)`; this is an explicit selection count, not an empty-model decision made by S2M.

4 Gaussian fitting with bayesreg

`Sequential2Means()` fits a Gaussian regression through `bayesreg` and applies S2M to all returned coefficient draws. Its priors are `ridge` (`rr`), `lasso`, `horseshoe` (`hs`) and `horseshoe+` (`hs+`). The intercept is excluded.

```
set.seed(12)
X <- matrix(rnorm(120), 40, 3)
colnames(X) <- c("x1", "x2", "x3")
y <- 3*X[,1] + rnorm(40)
fit <- Sequential2Means(X, y, b.i=c(.1, .5), prior="ridge",
                        n.samples=100, burnin=100)
dim(fit$Beta)
S2MVarSelectionV1(fit, fit$H.b.i[1])
```

Use adequate sampling and check convergence before interpreting the results. For draws from a separate `bayesreg` fit, transpose its coefficient matrix:

```
posterior <- bayesreg::bayesreg(y ~ ., data.frame(X, y), model="gaussian",
                               prior="horseshoe", n.samples=5000, burnin=2000, n.cores=1)
Beta <- t(posterior$beta)
result <- Sequential2MeansBeta(Beta, lower=.1, upper=.5, l=5)
```

5 Gaussian fitting and prediction

The native samplers return coefficient draws, intercept draws and residual variances with separate iteration and chain axes. `FitRidge()` draws independent posterior samples. `FitHorseshoe()` and `FitSpikeSlab()` use iterative sampling. They center X and Y without standardizing predictor columns. Their priors are specified in `?FitGaussian` and differ from the `bayesreg` defaults.

```
X.native <- cbind(x1=c(-2,-1,0,1,2,0), x2=c(0,1,-1,0,1,-1))
y.native <- c(-3,-1,0,1,3,.5)
native <- FitRidge(X.native,y.native,draws=1000,chains=2,seed=8)
summary(native)
#>           mean           sd          lower          upper
#> x1 1.26492510 0.2796437 0.7016238 1.7970364
#> x2 0.05655154 0.4312693 -0.7921984 0.9167098
dim(native$beta)
#> [1] 1000    2    2
Beta.native <- PosteriorDraws(native)
dim(Beta.native)
#> [1] 2000    2
```

`summary()` reports coefficient means, standard deviations and equal-tail intervals. `PosteriorDraws()` pools the iteration and chain axes in its returned matrix; the fitted object retains separate chains.

```
PlotCoefficients(native)
```

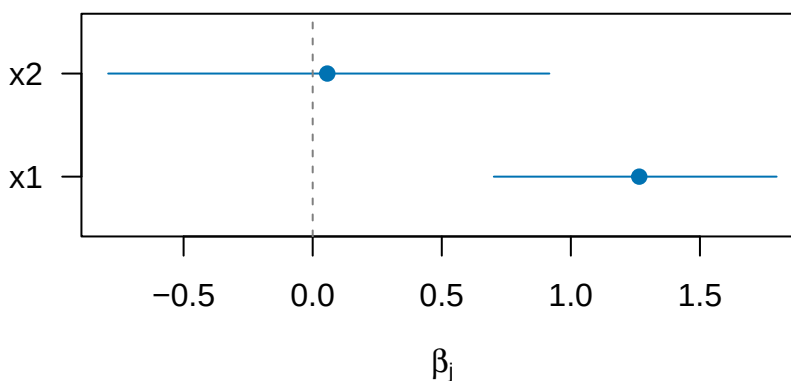


Figure 2: Posterior coefficient means and 95 percent equal-tail intervals in the ridge example. The dashed line marks zero.

Predictors for new observations must have the same column definitions and order. Prediction reuses posterior draws; it does not fit another model.

```
predict(native,X.native[1:2,,drop=FALSE])
#> $mean
#> [1] -2.444100 -1.122624
#>
#> $lower
#> [1] -3.759183 -2.416262
#>
#> $upper
#> [1] -1.138029 0.213716
#>
#> $interval
#> [1] 0.95
```

```

#>
#> $kind
#> [1] "mean"
predict(native,X.native[1:2,,drop=FALSE],include.noise=TRUE,seed=9)
#> $mean
#> [1] -2.444100 -1.122624
#>
#> $lower
#> [1] -4.791060 -3.561493
#>
#> $upper
#> [1] -0.2375214  1.1583263
#>
#> $interval
#> [1] 0.95
#>
#> $kind
#> [1] "observation"

```

The first call gives pointwise intervals for the conditional mean. The second adds observation noise for posterior predictive intervals. Both report the same posterior conditional mean.

```

if (requireNamespace("posterior",quietly=TRUE)) ChainDiagnostics(native)
#>               rhat ess.bulk ess.tail  mcse.mean          status
#> beta[x1]  1.0010304 1921.556 2007.396 0.006382279 inspect diagnostics
#> beta[x2]  1.0000063 1946.912 1866.274 0.009754355 inspect diagnostics
#> intercept 1.0008334 1941.033 1982.277 0.008498700 inspect diagnostics
#> sigma2    0.9995743 1936.265 1966.351 0.012398796 inspect diagnostics

```

`ChainDiagnostics()` uses the `posterior` package for rank R-hat, bulk and tail effective sample sizes, and the Monte Carlo standard error of the mean. It requires explicit chain identities and does not reconstruct chains from pooled draws. A single chain has no R-hat; constant variables have unavailable diagnostics. `PlotTrace()` and `PlotAutocorrelation()` display chains separately. Finite diagnostics alone do not establish convergence.

The trace and autocorrelation plots below use the saved horseshoe example with five observations and four chains. Each chain retains 3000 draws after 1000 burn-in iterations. The trace shows the first 300 retained draws; the within-chain autocorrelations use all 3000.

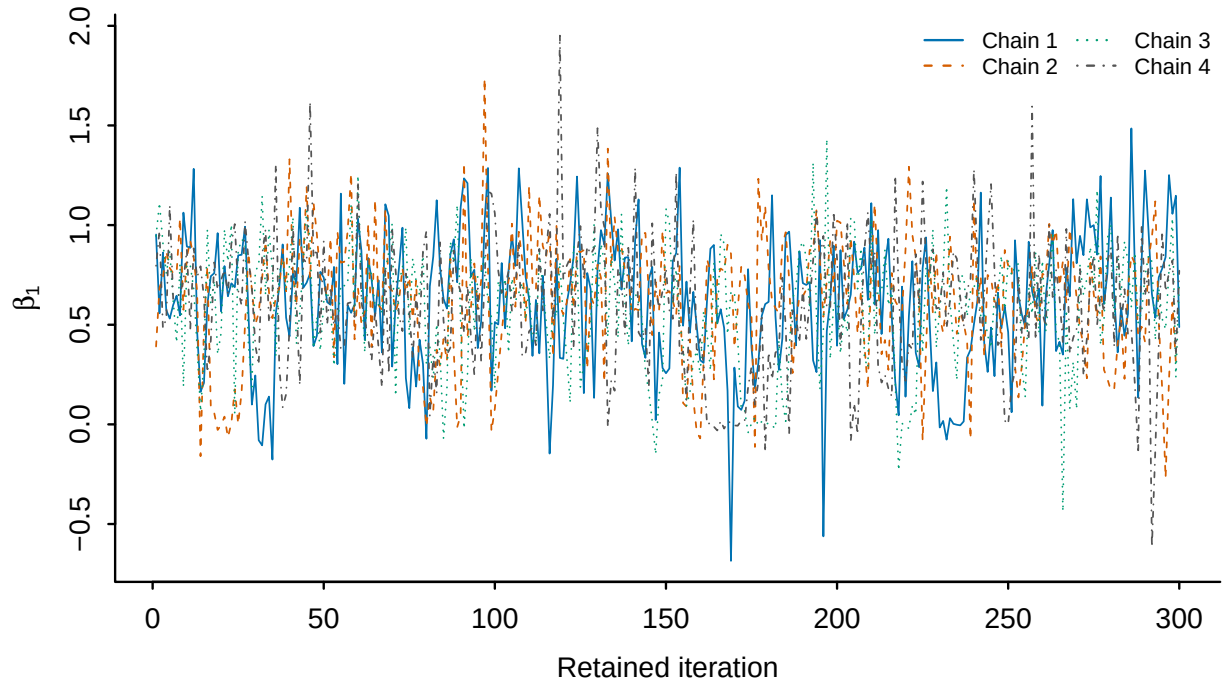


Figure 3: First 300 retained coefficient draws in four horseshoe chains. The short excerpt illustrates the display; it does not establish convergence.

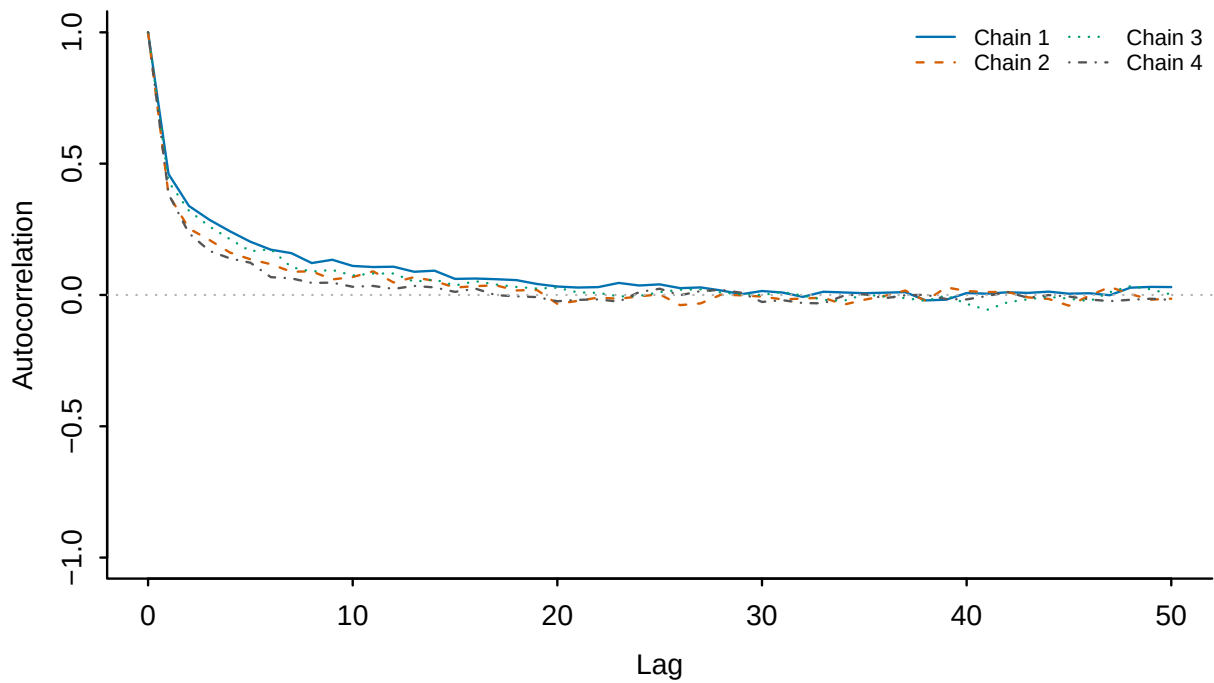


Figure 4: Within-chain sample autocorrelations for the same coefficient, using 3000 retained draws per chain.

To summarize which coefficients are included, fit a spike-and-slab model:

```

native.ss <- FitSpikeSlab(X.native,y.native,draws=200,burnin=100,chains=2,seed=4)
summary(SpikeSlabSelection(PosteriorDraws(native.ss)))
#>      inclusion.frequency      MPM
#> x1              0.9075    TRUE
#> x2              0.0575    FALSE

PlotInclusion(native.ss)

```

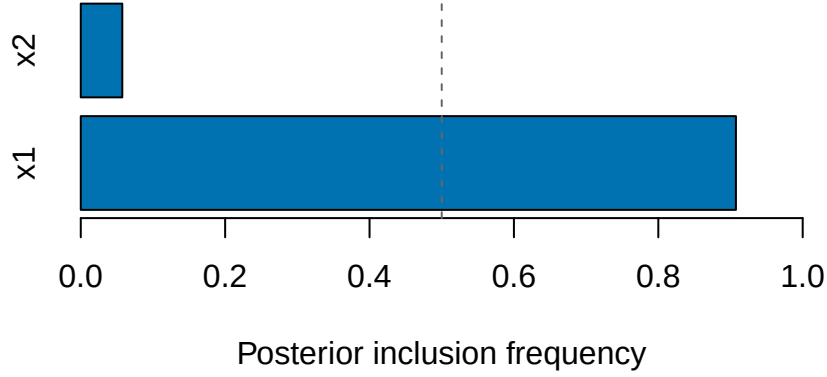


Figure 5: Inclusion frequencies from the short spike-and-slab example, with the median probability model cutoff at 0.5. The short run does not establish convergence.

6 SAVS and empty selections

With default handling, S2M leaves at least one signal. For a separate rule that can return no variables, VsusP provides the signal adaptive variable selector (SAVS) of Ray and Bhattacharya (2018). Let $\hat{\beta}_j$ be a supplied coefficient estimate and X_j its predictor column. With $a_+ = \max(a, 0)$, their closed form is

$$\hat{\beta}_j^* = \text{sign}(\hat{\beta}_j) \|X_j\|^{-2} \left(|\hat{\beta}_j| \|X_j\|^2 - |\hat{\beta}_j|^{-2} \right)_+.$$

An exactly zero estimate gives zero by continuity. A coefficient survives only when $|\hat{\beta}_j|^3 \|X_j\|^2 > 1$; equality gives zero. Every coefficient can therefore be removed. The implementation protects against overflow and uses an explicit roundoff convention for numerically ambiguous extreme-scale scores.

```

SAVS(diag(2),c(.1,-.1))$selected
#> integer(0)
SAVS(diag(2),c(2,0))$coefficients
#> [1] 1.75 0.00
SAVS(X.native,coef(native))$selected
#> [1] 1

```

Use the same predictor matrix and units that define the estimates, and exclude the intercept. No scaling occurs inside `SAVS()`. Multiplying a predictor by a factor and dividing its coefficient by that factor preserves the linear predictor but changes the SAVS decision score. Standardize predictors before fitting if selection on a common predictor scale is wanted. The closed form is not the full penalized optimizer for correlated predictors and does not provide an automatic false-discovery guarantee.

7 DSS sparse prediction

To retain a short predictor list while approximating a fitted linear predictor, `DSS()` uses the adaptive-L1 criterion of Hahn and Carvalho (2015, equation 21). Let $\bar{\beta}$ be the posterior mean and let X contain the

prediction points. We minimize over the sparse vector γ :

$$\frac{1}{n} \|X\bar{\beta} - X\gamma\|_2^2 + \lambda \sum_{j:\bar{\beta}_j \neq 0} \frac{|\gamma_j|}{|\bar{\beta}_j|}, \quad \lambda \geq 0.$$

For $\lambda > 0$, entries with $\bar{\beta}_j = 0$ are constrained to zero. For $\lambda = 0$, the function returns $\bar{\beta}$. This is the paper's convex approximation to its variable-count penalty. The supplied X may differ from the original fitting design, but its columns must retain the same definitions and order. Keep the intercept separately.

```
beta.mean <- coef(native)
dss <- DSS(X.native,beta.mean,lambda=.1)
dss$selected
#> [1] 1
dss$converged
#> [1] TRUE
penalties <- c(0,.01,.1,1,10)
dss.path <- lapply(penalties,function(lambda)DSS(X.native,beta.mean,lambda))
stopifnot(all(vapply(dss.path,function(x)x$converged,logical(1))))
data.frame(lambda=penalties,
            selected=vapply(dss.path,function(x)length(x$selected),integer(1)),
            mse=vapply(dss.path,function(x)x$mse,numeric(1)))
#>   lambda selected      mse
#> 1    0.00        2 0.00000000
#> 2    0.01        1 0.002088125
#> 3    0.10        1 0.003016229
#> 4    1.00        1 0.095826670
#> 5   10.00        0 2.692702372
```

The penalty λ controls the trade-off between sparsity and approximation error; the function does not choose it automatically. The reported `mse` compares $X\bar{\beta}$ with $X\gamma$ at the supplied points. It is not an estimate of error on new outcomes. Coordinate descent reports its relative optimality error as `kkt.error`; a warning and `converged=FALSE` identify an unmet tolerance. Dependent columns can give different selected sets with the same optimal fitted values. No inclusion probabilities or false-discovery control are claimed.

8 Clustering and tuning sensitivity

The default S2M path retains `kmeans` and the legacy small-set convention. Two optional choices make computational decisions explicit:

- `clustering="exact"` minimizes the two-cluster within-group sum of squares over distinct adjacent cuts in sorted coefficient magnitudes. It uses no random initialization. Extreme numerical ranges that erase distinctions between inputs are rejected.
- `small.set="refine"` splits a distinct recursive pair and applies the same strict gap comparison. The default "legacy" retains that pair. This choice can change the final signal count.

```
pair <- matrix(c(0,.125,10),nrow=1)
Sequential2MeansBeta(pair,.1,.1,1,"exact","legacy")$H.b.i
#> [1] 1
Sequential2MeansBeta(pair,.1,.1,1,"exact","refine")$H.b.i
#> [1] 2
```

For a threshold grid, `S2MSensitivity()` reports modal counts, their frequencies and selected-set overlap. Jaccard similarity is the size of the intersection of adjacent selected sets divided by the size of their union.

```
path <- S2MSensitivity(Beta,c(.1,.5),clustering="exact",small.set="refine")
data.frame(b=path$b.i,H=path$H.b.i,mode=path$mode.proportion,
```

```

overlap=path$jaccard.previous)
#>      b H mode overlap
#> 1 0.1 4      1      NA
#> 2 0.5 2      1      0.5

```

Count frequencies are not inclusion probabilities. An exact clustering optimum or stable selected set does not establish that the selected variables are correct. Neither option chooses an optimal threshold.

9 Spike-and-slab posterior draws

A spike-and-slab fit gives an exact zero for each excluded coefficient. `SpikeSlabSelection()` uses these zeros to estimate inclusion probabilities:

```

Beta.ss <- rbind(c(1,0,0), c(2,0,0), c(0,-1,0), c(0,0,0))
colnames(Beta.ss) <- c("x1", "x2", "x3")
ss <- SpikeSlabSelection(Beta.ss)
ss$inclusion.probability
#>   x1   x2   x3
#> 0.50 0.25 0.00
ss$MPM
#> [1] 1
ss$HPPM
#> [1] 1

```

These fixed draws illustrate the input. Replace them with posterior draws from a spike-and-slab fit, excluding the intercept. The median probability model (MPM) includes coefficients with inclusion frequency at least 0.5. The empirical highest posterior probability model (HPPM) is the most frequent complete model visited by the supplied draws. It need not be the most probable model over all possible models. Tied model frequencies favor the first visited model; all tied model indices are returned. Small continuous-prior coefficients are not spikes.

`TwoMeansBeta()` implements the basic 2-M rule: for each draw, take the smaller of the two cluster sizes, then use the modal count and median absolute coefficients to select variables. This differs from S2M's lower-mean noise set.

```

TwoMeansBeta(Beta.ss)$selected
#> [1] 1
Sequential2MeansBeta(Beta.ss,.1,.5,2,clustering="exact",zero.draws="empty")
#> threshold count
#>      0.1      1
#>      0.5      1

```

An entirely zero draw has no two-means split. `TwoMeansBeta()` assigns it zero signals. S2M does so only with `zero.draws="empty"`; its default still rejects such draws. These are explicit boundary extensions. Nonzero constant-magnitude draws remain invalid. Median-ranking ties can select a coefficient never included in any draw when all posterior medians are zero. Use the inclusion summaries when inclusion probability is the quantity of interest.

10 Selection performance in a small pilot

A reproducible pilot used 60 Gaussian datasets, with 10 datasets in each of six conditions. Each had $n = 80$, $p = 10$ and residual standard deviation one. The first two coefficients were $(0, 0)$, $(0.25, -0.25)$ or $(1, -1)$; the others were zero. Predictors had AR(1) correlation $\rho = 0$ or 0.8 and were centered and standardized by their sample standard deviations before generating responses. Horseshoe fits used one worker, 1,000 retained draws, burn-in 500 and thinning 5.

Effect	Correlation	Datasets	Empty	FP	FN	Exact
0.00	0.0	10	1.0	0.0	0.0	1.0
0.25	0.0	10	0.5	0.0	1.5	0.0
1.00	0.0	10	0.0	0.0	0.0	1.0
0.00	0.8	10	1.0	0.0	0.0	1.0
0.25	0.8	10	1.0	0.0	2.0	0.0
1.00	0.8	10	0.0	0.3	0.0	0.8

Here Empty and Exact are proportions of datasets with an empty selection or exactly the true variable set. FP and FN are mean counts of false positives and false negatives. There were no failed fits or recorded warnings. SAVS returned an empty set in all 20 null datasets, but it missed weak signals, especially in the correlated opposite-sign case. The pilot is small: a proportion of 10/10 has a 95% Wilson interval of about 0.72 to 1.00.

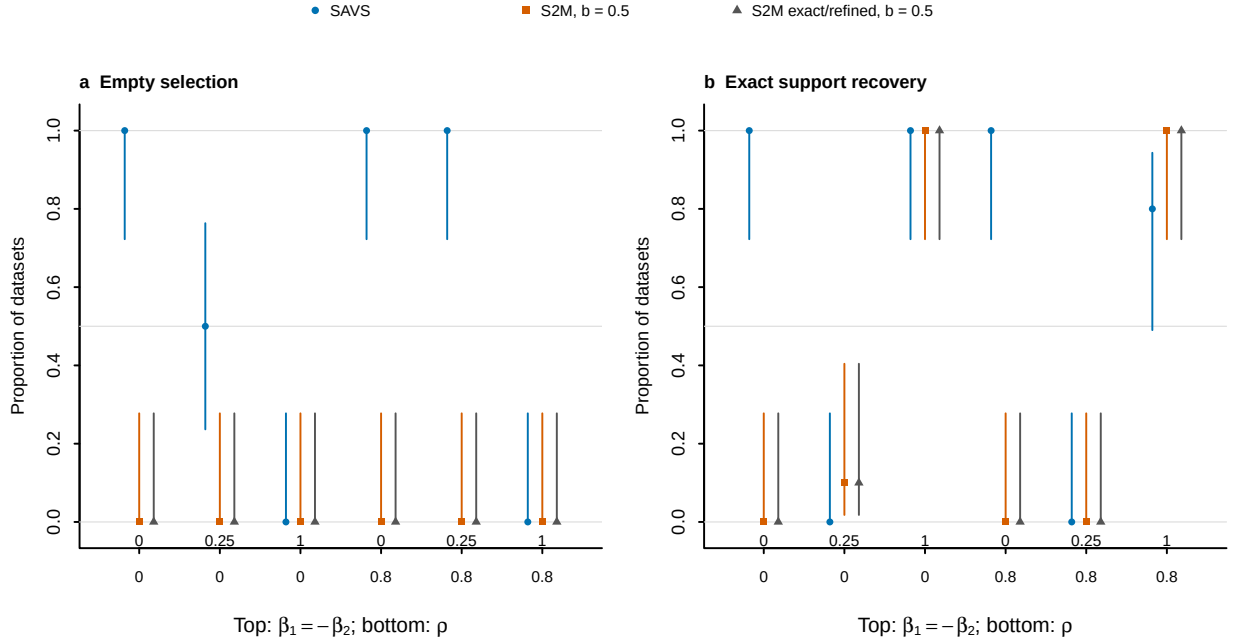


Figure 6: Empty selections and exact-support recovery across 10 datasets per condition, with 95 percent Wilson intervals. Labels give the first coefficient (the second is its negative) and predictor correlation. S2M thresholds are fixed, not optimized.

Paired predictor-unit changes altered some SAVS selections while preserving the fitted linear predictor. Undoing those changes restored the original selections in every case. Selection based on the first and second posterior halves differed in 2 of 60 cases. This comparison is an instability screen, not a convergence check. The pilot does not establish false-discovery control or general accuracy. The complete outcomes and runner are installed under `validation/`.

11 Fixed-posterior checks

To check the native horseshoe and spike-and-slab fits, we compare their sampled posterior summaries with independent numerical integration for the same five-observation, one-predictor input. Each fit uses four chains, 1000 burn-in iterations and 3000 retained draws per chain. We compare coefficient means and mean

residual variance; the spike-and-slab check also compares inclusion probability. These fixed inputs test the fitted posterior calculations rather than selection accuracy across repeated datasets.

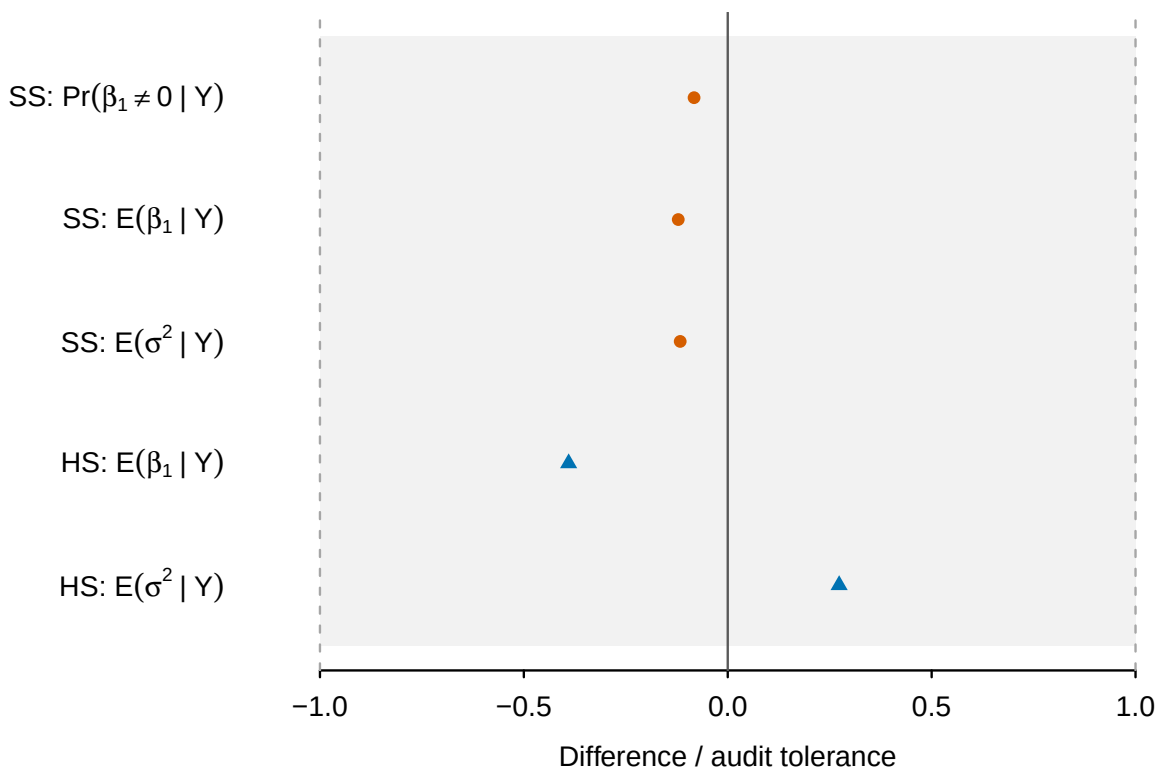


Figure 7: Sampled-minus-integrated summaries divided by the comparison tolerance: the larger of 0.01 and six Monte Carlo standard errors from 100-draw batches. HS denotes horseshoe and SS spike-and-slab. Shading marks the comparison criterion, not a confidence interval.

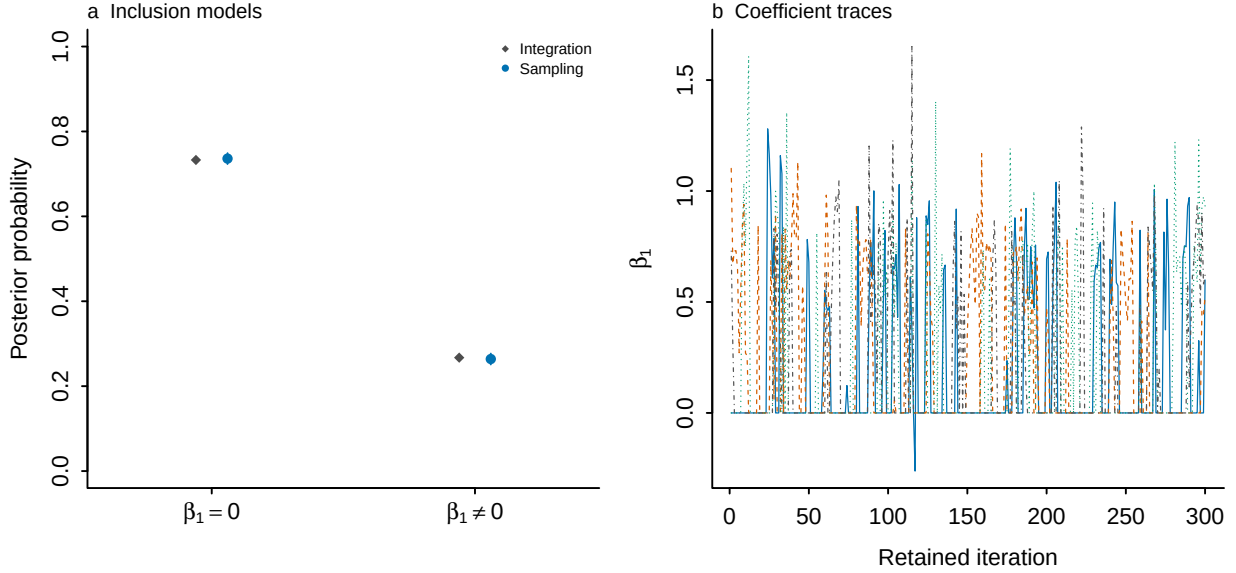


Figure 8: Spike-and-slab inclusion-state probabilities from numerical integration and sampling, with plus or minus 1.96 Monte Carlo standard errors. The trace excerpt shows the first 300 retained draws per chain; all 3000 enter the summaries.

The displayed differences meet the stated comparison tolerances. Agreement in this small example does not establish reliable mixing in high-dimensional models or interval coverage across repeated datasets. The numerical inputs and figure source are supplied in `doc-data/`.

12 Interpretation and limits

The threshold b is on the coefficient scale. Scaling predictors changes its meaning, so inspect the tuning path and sensitivity to scaling and b . `kmeans` initialization can affect results; set a random seed for reproducibility.

S2M requires at least two coefficients and at least two distinct absolute values in each initial draw. An initial pair of distinct values gives one noise coefficient and one signal. Constant-magnitude draws are rejected because they have no lower-mean cluster. After an initial split, an indivisible lower-mean candidate is retained without another split.

By default, every accepted draw leaves at least one signal, so S2M cannot automatically select an empty model. Its modal counts and selected variables are clustering summaries. They are not posterior inclusion probabilities, significance tests or a guarantee of false-positive control.

13 References

- Hahn, P. R. and Carvalho, C. M. (2015). Decoupling Shrinkage and Selection in Bayesian Linear Models: A Posterior Summary Perspective. *Journal of the American Statistical Association*, **110**(509), 435–448. <https://doi.org/10.1080/01621459.2014.993077>.
- Li, H. and Pati, D. (2017). Variable selection using shrinkage priors. *Computational Statistics & Data Analysis*, **107**, 107–119. <https://doi.org/10.1016/j.csda.2016.10.008>.
- Makalic, E. and Schmidt, D. F. (2016). High-dimensional Bayesian regularised regression with the BayesReg package. arXiv:1611.06649. <https://doi.org/10.48550/arXiv.1611.06649>.
- Ray, P. and Bhattacharya, A. (2018). Signal Adaptive Variable Selector for the Horseshoe Prior. arXiv:1810.09004. <https://arxiv.org/abs/1810.09004>.

Wang, H. and Song, M. (2011). Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming. *The R Journal*, 3(2), 29–33. <https://doi.org/10.32614/RJ-2011-015>.