Reproduction of the Hamming Ball Sampler

Ruoran Huang, Shuyu Guo

December 2024

1 Introduction

Sampling from high-dimensional discrete-valued spaces often requires efficient sampling methods due to the "curse of dimensionality" and complex dependencies among variables. Traditional Markov Chain Monte Carlo (MCMC) methods like Gibbs sampling or block-conditional Gibbs sampling can encounter issues like posterior intractability and inability to escape from local modes in the posterior distribution with strongly correlated variables.

The Hamming Ball Sampler (HBS) paper by Titsias and Yau [1] sets out to target these limitations by employing auxiliary variables that allow sampling from a local "Hamming ball" around the current state in the discrete space. A Hamming ball counts the total number of states that differ from the current state within a Hamming distance, the position at which the corresponding symbols are different. By focusing updates within this local neighborhood, HBS can effectively improve mixing and computational efficiency.

2 Methodology

Given observations $y = \{y_1, \ldots, y_N\}$, a latent discrete-valued variable X, and model parameters, the joint distribution is

$$P(y, X, \theta) = \left(\prod_{i=1}^{N} P(y_i \mid X, \theta) \right) P(X, \theta),$$

where $X = \{x_{ij}\}, x_{ij} \in \{1, \ldots, S\}$. The posterior distribution of interest is:

$$P(X, \theta | y) \propto P(y, X, \theta).$$

Key Idea of HBS

The Hamming Ball Sampler introduces an auxiliary variable U that defines a **Hamming Ball $\boldsymbol{H_m(X)}$**, to transform the problem into a more tractable space:

$$P(y, X, \theta, U) = P(y, X, \theta)P(U|X),$$

where $P(U \mid X)$ is uniform over $H_m(X)$:

$$P(U|X) = \frac{1}{Z_m}\mathbb{I}(U \in H_m(X)),$$

where Z_m is the cardinality of $\mathcal{H}_m(X)$. Here, the Hamming Ball is defined as:

$$H_m(X) = \{U : d(X, U) \leqslant m\},$$

where $d(X, U)$ is the **Hamming Distance** $\sum_i \mathbb{I}(x_i \neq u_i)$, where the pairs $(\mathbf{u}_i, \mathbf{x}_i)$ denote non-overlapping subsets of corresponding entries in (U, X) such that $\bigcup_{i=1}^{P} u_i = U$ and $\bigcup_{i=1}^{P} x_i = X$. The parameter m denotes the maximal distance or radius of each individual Hamming Ball set (the number of bits we can change at once).

Specifically, we divide X into P blocks with K elements in each block, then

$$\mathcal{H}_m(X) = \{U : d(u_i, x_i) \leqslant m, i = 1, ..., P\}$$

$$\Rightarrow P(U|X) = \prod_{i=1}^{P} \frac{1}{Z_{i,m}}\mathbb{I}(d(u_i, x_i) \leqslant m),$$

where $Z_{i,m}$ is the volume of the individual Hamming ball set and $Z_{i,m} = M = \sum_{j=1}^{m}(S-1)^j\binom{K}{j}$.

Sampling Steps

1. **Sample** U: Sample the auxiliary variable U uniformly from the Hamming Ball $H_m(X)$:
$$U \sim P(U \mid X) = \frac{\mathbb{I}(U \in H_m(X))}{Z_m}, \quad Z_m = |H_m(X)|.$$
2. **Sample** X **and** θ: Given U, sample X and θ from their conditional posterior:
$$(X, \theta) \sim P(X, \theta \mid y, U).$$
Especially,

- X is sampled conditioning on θ, U and y:

$$P(X \mid \theta, U, y) \propto P(y, X, \theta)\mathbb{I}(X \in H_m(U)).$$

  This is tractable because,

$$\begin{aligned} P(X \mid \theta, y, U) &= \frac{P(X, \theta, y)P(U|X)}{P(\theta, y, U)} = \frac{P(X, \theta, y)\mathbb{I}(U \in H_m(X))}{P(\theta, y, U)Z_m} \\ &= \frac{P(X, \theta, y)\mathbb{I}(X \in H_m(U))}{P(\theta, y, U)Z_m} \quad \text{(according to symmetry)} \\ &\propto P(X, \theta, y)\mathbb{I}(X \in H_m(U)) \end{aligned}$$

  which means to sample X from the space $H_m(U)$ with weight $P(X, \theta, y)$.

- θ is sampled from

$$\theta \sim P(\theta \mid X, y)$$

A visualization of sampling procedure is given by the author [1] as below:

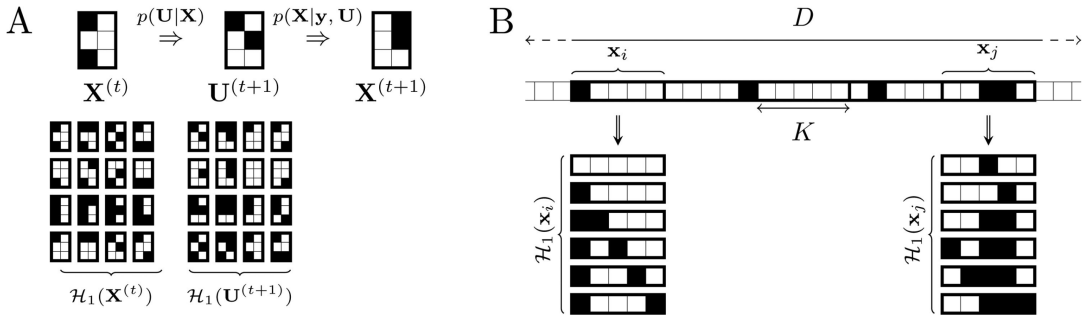


Figure 1: Hamming Ball Sampler Illustration (Titsias, 2017)

Pseudocode for HBS

Algorithm 1 Hamming Ball Sampler (HBS)

Require: Observations $\boldsymbol{y}$, initial inclusion vector $X^{(0)}$, number of iterations T, block size D/P, Hamming ball radius m.

Ensure: Samples of X and θ.

Initialize: Set $t = 0$, $X^{(0)}$.

for $t = 1$ to T **do**

  Randomly partition $X^{(t-1)}$ into P blocks $\{x_1, x_2, \ldots, x_P\}$ of size D/P.

  for $i = 1$ to P **do**

    Sample auxiliary variable $u_i^{(t)}$:

$$u_i^{(t+1)} \sim P(u_i \mid x_i^{(t)}) = \frac{\mathbb{I}(u_i \in H_m(x_i^{(t)}))}{\sum_{u_i} \mathbb{I}(u_i \in H_m(x_i^{(t)}))},$$

    where $H_m(x_i^{(t)})$ is the Hamming ball of radius m centered at $x_i^{(t)}$.

    Sample updated block $x_i^{(t)}$:

$$x_i^{(t+1)} \sim \frac{P(x_i, x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \ldots, x_P^{(t)}, \theta^{(t)}, \boldsymbol{y}) \mathbb{I}(x_i \in H_m(u_i^{(t+1)}))}{\sum_{x_i} P(x_i, x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \ldots, x_P^{(t)}, \theta^{(t)}, \boldsymbol{y}) \mathbb{I}(x_i \in H_m(u_i^{(t+1)}))}.$$

  end for

  Sample global parameter $\theta^{(t)}$:

$$\theta^{(t+1)} \sim P(\theta \mid X^{(t+1)}, \boldsymbol{y}).$$

end for

3 Computational Time Complexity

The maximum Hamming distance between U and $X^{(t)}$ is mP, where m is the radius of the Hamming ball and P is the number of blocks. Similarly, the Hamming distance between U and $X^{(t+1)}$ is also constrained by mP. Thus, overall $X^{(t+1)}$ can differ at most $2mP$ elements from previous iteration. When the Hamming ball radius m equals to the block size K, the Hamming ball sampler reduces to the standard block Gibbs sampling.

Assuming $p(X|\theta, U, y)$ factorizes across P blocks of size K, the computation time complexity of Hamming ball sampler scales by $O(MP)$ where M is the number of configurations within a single Hamming ball, given by:

$$M = \sum_{j=0}^{m} (S-1)^j \binom{K}{j}.$$

Here:

- S is the number of discrete states for each variable.
- $\binom{K}{j}$ is the binomial coefficient, representing the number of ways to select j positions out of K.
- $(S-1)^j$ corresponds to the number of alternative values for the selected j positions.

When $m = K$, the Hamming Ball Sampler simplifies to Block Gibbs Sampling, and the computational complexity becomes $O(S^K P)$, where:

$$S^K = \sum_{j=0}^{K} (S-1)^j \binom{K}{j}.$$

This demonstrates that the Hamming Ball Sampler is a flexible extension of Block Gibbs Sampling, allowing the user to control computational cost by varying the radius m.

4 Simulation Study

Model Setup

The sparse linear regression model is used to perform variable selection in high-dimensional settings. The goal is to identify which covariates contribute to explaining the response variable $\boldsymbol{y}$.

- **Response Variable** ($\boldsymbol{y}$): $N \times 1$ vector of observed responses normalized to have zero mean.
- **Design Matrix** ($\boldsymbol{Z}$): $N \times D$ matrix containing covariates for N observations and D predictors.
- **Latent Inclusion Vector** ($\boldsymbol{X}$): A binary vector ($D \times 1$) where each element $x_d \in \{0, 1\}$ indicates whether the d-th covariate is included in the regression model.

The observed responses are modeled as:

$$\boldsymbol{y} = \boldsymbol{Z}_X \boldsymbol{\beta}_X + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_N),$$

where:

- $\boldsymbol{Z}_X$: Submatrix of $\boldsymbol{Z}$, corresponding to predictors with $x_d = 1$.
- $\boldsymbol{\beta}_X$: Regression coefficients for selected predictors ($D_X \times 1$).
- $\boldsymbol{\eta}$: Gaussian noise, with variance σ^2.

Prior Distributions

1. **Inclusion Probability**: Each element of the inclusion vector $\boldsymbol{X}$ follows a Bernoulli distribution, with the inclusion probability π_0 following a Beta prior:

$$\pi_0 \sim \text{Beta}(\alpha_{\pi_0}, b_{\pi_0}).$$

2. **Regression Coefficients and Noise Variance**: Conjugate g-prior on $\boldsymbol{\beta}_X$

$$p(\boldsymbol{\beta}_X, \sigma^2 \mid \boldsymbol{X}) = \mathcal{N}(\boldsymbol{\beta}_X \mid \mathbf{0}, g(\boldsymbol{Z}_X^\top \boldsymbol{Z}_X)^{-1}) \text{InvGa}(\sigma^2 \mid \alpha_\sigma, b_\sigma),$$

where:

- g: Scales the prior covariance matrix, controlling the strength of the prior on $\boldsymbol{\beta}_X$.
- $\text{InvGa}(\sigma^2 \mid \alpha_\sigma, b_\sigma)$: The inverse-gamma prior on the noise variance σ^2.

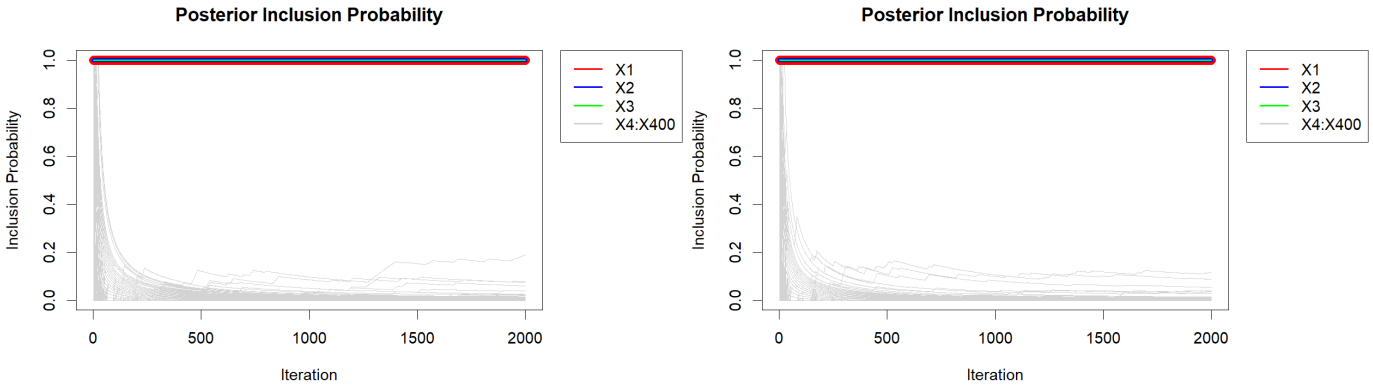
Experimental Setup

The simulation was conducted with the following settings:

- **Dataset Size**: $N = 100$ responses and $D = 400$ potential covariates.
- **Independent Design Matrix $\boldsymbol{Z}_{\text{ind}}$**: In the independent case, the columns of $\mathbf{Z}$ are uncorrelated. The entries of $\mathbf{Z}$ are sampled independently from standard normal:
$$z_{ij} \sim \mathcal{N}(0, 1), \quad \forall i = 1, \ldots, 100, \; j = 1, \ldots, 400.$$
- **Dependent Design Matrix $\boldsymbol{Z}_{dep}$**: In the dependent cases, the columns of $\mathbf{Z}$ are correlated, with covariance matrix defined as:
$$\Sigma_{ij} = \begin{cases} 1, & \text{if } i = j \text{ (diagonal entries)}; \\ \rho, & \text{if } i \neq j \text{ (off-diagonal entries)}. \end{cases}$$
The design matrix $\mathbf{Z}_{\text{dep}}$ is generated by sampling from a multivariate normal distribution:
$$\mathbf{Z}_{\text{dep}} \sim \mathcal{N}(\mathbf{0}, \Sigma).$$
We consider two cases: $\rho = 0.5$ and $\rho = 0.7$.
- We draw η_i's independently from $\mathcal{N}(0, 1)$. and generate Y with the first three covariates:
$$Y_i = X_{i1} + X_{i2} + X_{i3} + \eta_i$$
- We randomly segment X into 10 blocks at each iteration, and each segment is updated as described in pseudo code 1.
- The initial X_i's are drawn from $\{0, 1\}$ with probability $\{0.9, 0.1\}$.

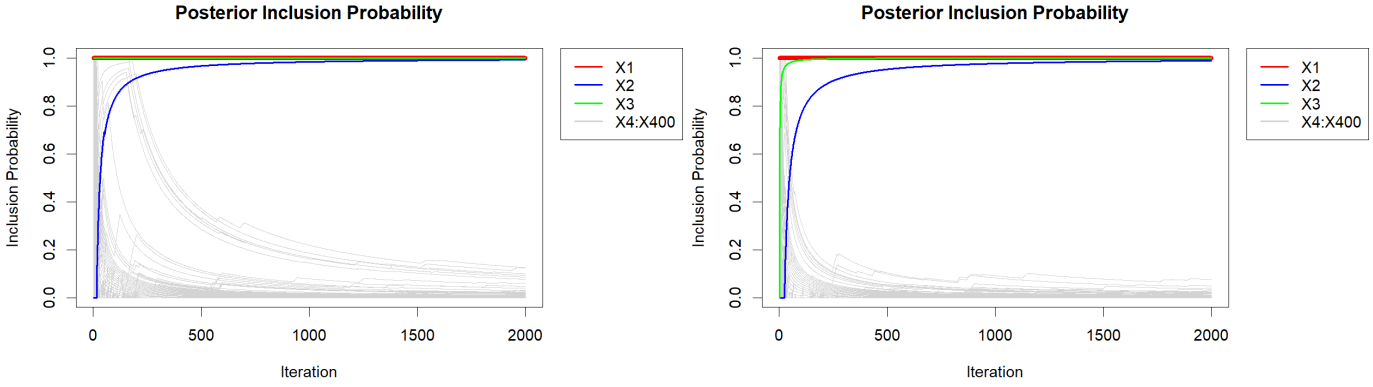
Results

We set the hamming ball radius $m \in \{1, 2\}$ and run 2000 iterations for each data set. Figure 2 shows the traces of posterior inclusion probability of each covariate, i.e. the cumulative proportion of $\{X_i = 1\}$, across iterations. Traces for X_1, X_2 and X_3 are in red, blue and green respectively. And we also plot the traces for other covariates which are not included in the true model in light grey.



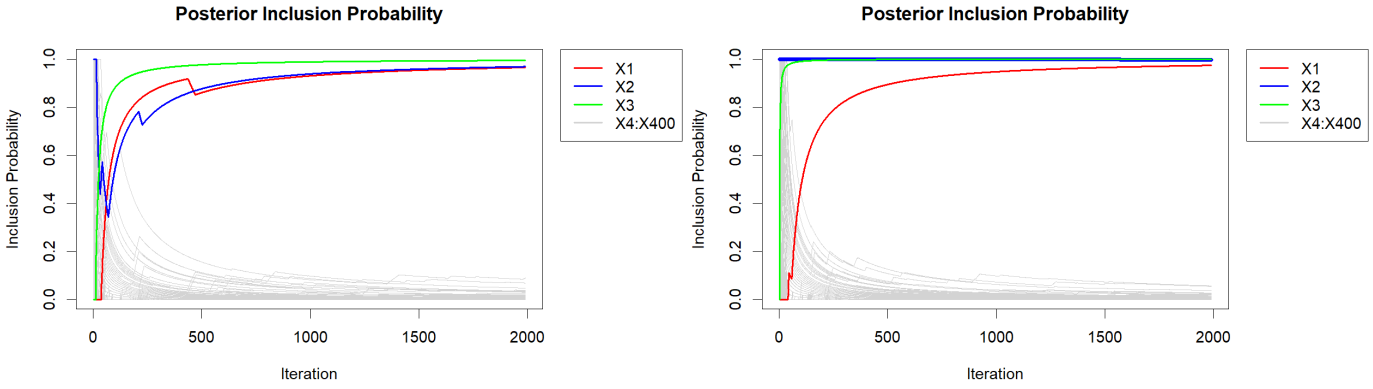
(a) Z_{ind}, $m = 1$

(b) Z_{ind}, $m = 2$



(c) $Z_{dep1}(\rho = 0.5)$, $m = 1$

(d) $Z_{dep1}(\rho = 0.5)$, $m = 2$



(e) $Z_{dep2}(\rho = 0.7)$, $m = 1$

(f) $Z_{dep2}(\rho = 0.7)$, $m = 2$

Figure 2: Posterior Inclusion Probability of Covariates.

- When the the columns of the design matrix are independent (or slightly correlated), the three true covariates are detected right after the first iteration and are consistently included. And for those excluded covariates, their inclusion probabilities drop sharply and converge to zero. In such case, there is little difference between choosing $m = 1$ and $m = 2$. Thus, $m = 1$ is a better choice as it reduces computation complexity.

- A larger m allows the sampler to explore more candidates within one step and thus reduce the risk of stuck in a local mode. Therefore, as the correlations between columns become stronger, choosing a larger hamming ball radius enables the sampler to transition more efficiently and yields a faster convergence rate. We can observe that in the last row where $\rho = 0.7$, the convergence rate is significantly improved by choosing $m = 2$ instead of 1.

5 Strengths and Limitations

The Hamming Ball Sampler (HBS) has several strengths that make it a powerful tool. First, it enables efficient local exploration by restricting computations to localized neighborhoods defined by the Hamming ball, which significantly improves scalability. This feature is particularly beneficial for sparse models where the computational burden can otherwise be overwhelming. Besides, HBS provides improved mixing times by leveraging auxiliary variables to overcome local mode trapping and enable better exploration of the state space.

The Hamming Ball has to be relatively small, comparing to the whole space. So, the radius of the Hamming ball has to be carefully chosen to balance computational cost and exploration efficiency.
The size of the Hamming ball grows combinatorially with M and P. For small M, the complexity is manageable because the number of configurations in $\mathcal{H}_m(X)$ is much smaller than the total number of possible configurations of X. This gives faster mixing but can limit the exploration of the space. For large M, the size of $\mathcal{H}_m(X)$ grows rapidly and the computational cost is high.

Hamming ball sampler may require careful design and tuning for specific models.

6 Extensions and Conclusion

Some dynamic adjustments enable the Hamming Ball Sampler to be a more flexible sampling method:

- **Varying Hamming Distances:** Instead of a fixed radius m, the sampler can dynamically adjust the Hamming distance during the sampling process. Then, the conditional distribution over U is now uniform on the generalized Hamming ball:

  $$\mathcal{H}_m(X) = \{U : d(u_i, x_i) \leqslant m_i,\ i = 1, \ldots, P\},$$

  where $\mathbf{m} = (m_1, \ldots, m_P)$ denotes the set of maximal distances for each subset of variables. Using a varying hamming distance allows the algorithm to focus computational effort where it is most needed.

- **Non-Uniform Auxiliary Distribution:** We can use a more complex form for the auxiliary conditional distribution $p(U|X)$. One generalization can be in this form:

  $$p(U \mid X) = \prod_{i=1}^{P} \frac{1}{Z_{i,m}} \exp(-\lambda d(u_i, x_i)) \mathbb{I}(u_i \in \mathcal{H}_m(x_i)),$$

  where the parameter $\lambda \geqslant 0$ controls the variance of the distribution. Higher values of λ concentrate the probability mass around the center X. This leads to a more targeted exploration on the nearby neighborhood.

Given so many details about the HBS sampling procedure and extensions, we should not treat it as a universal solution for speeding up MCMC algorithm. Instead, it is just a novel alternative to the tools we already have.

References

[1] Michalis K. Titsias and Christopher Yau. The hamming ball sampler. *Journal of the American Statistical Association*, 112(520):1598–1611, 2017. 1, 3