Mini project: Signal recovery using MCMC

Lucas Gruaz, Michael Hauri, Andrea Miele

June 18, 2024

Question 1: Optimizing over binary hypercube

1.1

If $\sigma = 0$, the noise vector is always 0. Hence, $y = X\theta$. Finding θ given (X, y) is equivalent to solving a system of m equations with d variables θ_i for $i = 1, ..., d$. This system is guaranteed to yield a solution as long as $m \geq d$. So the minimum number of measurements m required to recover θ with probability 1 is $m = d$. An algorithm for finding θ is the following:

- compute X^{-1}, the inverse of X.
- $\theta = X^{-1}y$

The computational cost of this procedure is dominated by the matrix inversion operation, which has a complexity of $\mathcal{O}(d^3)$.

1.2

We first compute the expectation of y given θ, X:

$$\mathbb{E}[y|X, \theta] = \mathbb{E}[X\theta + \xi|X, \theta] = X\theta + \mathbb{E}[\xi] = X\theta$$

The variance of y comes from ξ, so y follows a multivariate normal distribution $\mathcal{N}(X\theta, \sigma^2 I_m)$. Thus,

$$\text{Prob}\{y|\theta, X\} = \frac{1}{(2\pi\sigma^2)^{m/2}}\exp(-\frac{1}{2\sigma^2}\|y - X\theta\|^2)$$

which is the likelihood function we aim to maximize. Maximizing the likelihood is equivalent to minimizing the negative log-likelihood:

$$L(\theta) = \frac{m}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\|y - X\theta\|^2$$

hence the optimization problem is:

$$\hat{\theta} = \arg\max_{\theta\in\Theta}(\text{Prob}\{y|\theta, X\}) = \arg\min_{\theta\in\Theta}(L(\theta)) = \arg\min_{\theta\in\Theta}(\frac{m}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\|y - X\theta\|^2)$$

1.3

$$\pi_\beta(\theta) = \frac{1}{Z_\beta}e^{-\beta L(\theta)}$$

where Z_β is a normalization constant.

1.4

We design a Metropolis-Hastings (MH) algorithm on the state space Θ. The base chains has transition as follows: from a state θ, the next state is selected by choosing a random i in $1, ..., d$ and flipping θ_i, i.e

$$\psi_{u,v} = \begin{cases} \frac{1}{d}, & \text{if } |u-v| = 1 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The acceptance probabilities are defined as

$$a_{u,v} = \begin{cases} \min(1, \frac{\pi_\beta(v)\psi_{v,u}}{\pi_\beta(u)\psi_{u,v}}), & \text{if } |u-v| = 1 \\ 0, & \text{otherwise} \end{cases} = \begin{cases} \min(1, e^{-\beta(L(v)-L(u))}), & \text{if } |u-v| = 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $|u - v|$ is the Hamming distance between u and v. An improving proposed move from u to v such that $L(v) \leq L(u)$ will always be accepted, while a worsening move such that $L(v) > L(u)$ may be accepted with some probability (to help escape local minima). As β tends to infinity, the probability distribution becomes increasingly concentrated on the values of θ that minimize $L(\theta)$.

Simplification of the computation of the acceptance probability

We note that given a state u, the generated proposal v is equal to u on all indices except one index $k \in [1, ..., d]$. We can derive $X \cdot v$ from $X \cdot u$ as follows:

$$(Xv)_i = \begin{cases} (Xu)_i + X_{i,k} & \text{if } u_k = 0, v_k = 1 \\ (Xu)_i - X_{i,k} & \text{if } u_k = 1, v_k = 0 \end{cases} \tag{3}$$

which we can use to compute

$$L(v) - L(u) = \frac{1}{2\sigma^2}(\|y - Xv\|^2 - \|y - Xu\|^2) \tag{4}$$

without having to perform the matrix multiplication at each step. The computational complexity of obtaining Xv from Xu is $\mathcal{O}(m)$, computing $\|y - Xv\|^2$ is also $\mathcal{O}(m)$. Hence, the complexity of computing the acceptance probabilities is $\mathcal{O}(m)$.

1.5

The MH scheme was implemented. Figure 1 shows the results for $d = 2000$, total Markov chain steps $= 100d$, $\sigma = 1$. All values of β show similar results. Increasing the number of measurements m improves the quality of the solution. When $m \geq d$, the optimal solution is reliably found. Simulated Annealing results are shown in Figure 2. Overall, it performs similarly as MH.

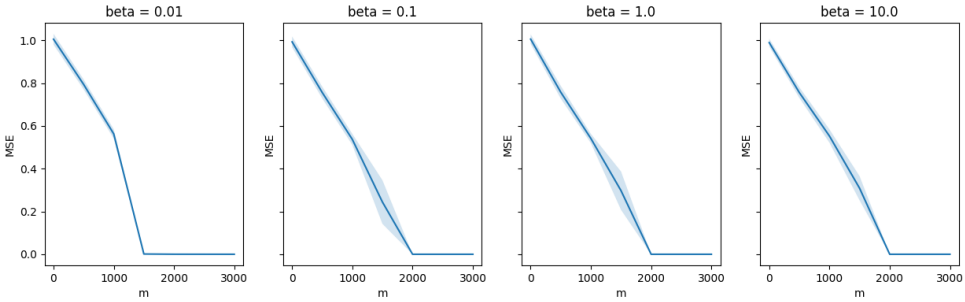


Figure 1: Metropolis-Hastings results for different values of β and m. In all case, the other parameters are set as $d = 2000$, total Markov chain steps $= 100d$, $\sigma = 1$. The mean over 30 realizations of X and θ is shown here, and the width of the shaded area depicts the standard deviation.

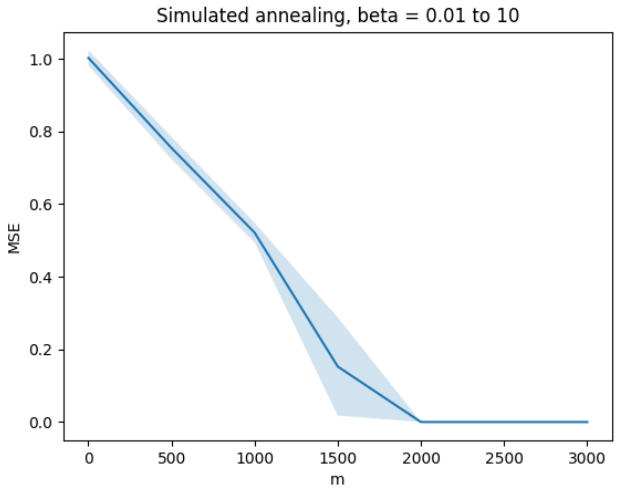


Figure 2: Simulated Annealing results for different values of m. β varies between 0.01 and 10, and the other parameters are set as $d = 2000$, total Markov chain steps $= 100d$, $\sigma = 1$. The mean over 30 realizations of X and θ is shown here, and the width of the shaded area depicts the standard deviation.

1.6

Figure 1 and 2 show the results. When we vary m from 1 to d, the performance increases. The minimum value of m/d required to reliably recover θ is 1.0 (i.e $m = d$).

Question 2: Recovering a sparse, binary signal

2.1

Description of the Base Chain

Let $\Theta = \{\theta \in \{0,1\}^d\}$ be the state space, where θ is a binary vector of length d. The base chain is defined by the following transition mechanism:

Given a current state $\theta^{(t)} \in \Theta$, the next state $\theta^{(t+1)}$ is obtained by:

1. Selecting two distinct indices $i, j \in \{1, 2, \ldots, d\}$ uniformly at random.
2. Swapping the values of $\theta_i^{(t)}$ and $\theta_j^{(t)}$. If $\theta_i^{(t)} \neq \theta_j^{(t)}$, this is equivalent to flipping both bits.

The transition probability from $\theta^{(t)}$ to $\theta^{(t+1)}$ is given by:

$$P(\theta^{(t+1)} = \theta' | \theta^{(t)} = \theta) = \begin{cases} \frac{1}{\binom{d}{2}} & \text{if } \theta' \text{ can be obtained by swapping any two bits in } \theta \\ 0 & \text{otherwise} \end{cases}$$

where $\binom{d}{2}$ represents the total number of ways to choose two distinct indices from d indices.

Characteristics of Sparse Binary Signals

A sparse binary signal means:

- Most elements in θ are zero.
- A small number, s, are ones.
- Probability of picking a non-zero bit is $\frac{s}{d}$, low when $s \ll d$.

Issue with the Base Chain in Sparse Settings

- **Inefficient Exploration:** Often results in no change, leading to poor exploration of the state space.
- **Low Probability of Meaningful Swaps:** Probability of choosing one one and one zero is small for $s \ll d$. **(*)**
- **Convergence Issues:** The chain may converge slowly if the initial state is far from the target distribution.

(*) Details:

The probability of one of the chosen coordinates being a one, and the other being a zero in a single swap operation, is calculated as:

$$P(\text{one one, one zero}) = \frac{s}{d} \times \frac{d-s}{d-1} + \frac{d-s}{d} \times \frac{s}{d-1}$$

It's important to note that as d becomes much larger than s, this probability becomes relatively small, making meaningful swaps less likely in each step of the MH algorithm.

2.2

Objective

The goal is to design a base chain that efficiently explores the space of sparse binary signals, enabling the MH algorithm to recover θ with a number of measurements m, proportional to $O(s \log d)$.

Proposed Modification

Let $\Theta = \{\theta \in \{0,1\}^d : \|\theta\|_0 = s\}$ be the state space, where $\|\theta\|_0$ denotes the number of non-zero components of θ, d is the dimension of the signal, and s is the sparsity (the number of ones in θ). The state θ is a binary vector of length d.

The modified base chain involves transitioning from a current state $\theta^{(t)}$ to a next state $\theta^{(t+1)}$ by swapping a 1 and a 0 in $\theta^{(t)}$. The proposal mechanism for generating a proposed state θ' from $\theta^{(t)}$ is as follows:

1. Select an index i uniformly at random such that $\theta_i^{(t)} = 1$.
2. Select an index j uniformly at random such that $\theta_j^{(t)} = 0$.
3. Swap the values of $\theta_i^{(t)}$ and $\theta_j^{(t)}$ to obtain θ'.

The transition probability of this process is defined as:

$$P(\theta^{(t+1)} = \theta' | \theta^{(t)} = \theta) = \begin{cases} \frac{1}{|\{i:\theta_i=1\}| \cdot |\{j:\theta_j=0\}|} & \text{if } \theta' \text{ can be obtained by swapping a 1 and a 0 in } \theta \\ 0 & \text{otherwise} \end{cases}$$

where $|\{i : \theta_i = 1\}|$ is the count of indices where θ has 1s, and $|\{j : \theta_j = 0\}|$ is the count of indices where θ has 0s. This assumes a uniform distribution over all possible swaps of a 1 and a 0 in the current state θ.

2.3

Given the modified base chain as described previously, we analyze its performance by plotting $\frac{1}{2s} E\|\hat{\theta} - \theta\|^2$ as a function of the number of measurements m. For this purpose, we considered dimensions d in the range of [2000, 5000], with sparsity $s = d/100$, total Markov chain steps equal to $100d$, and $\sigma = 1$.

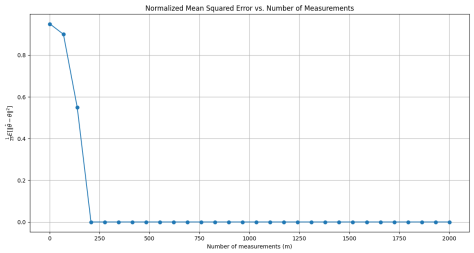
Characteristics of the Plots

Upon inspecting the plots, we should focus on the following characteristics:

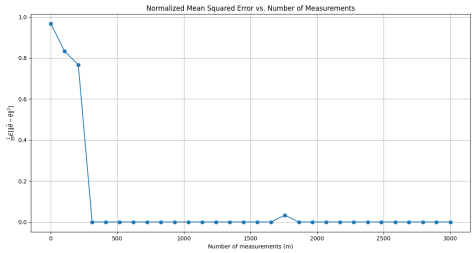
- The rate at which NMSE decreases as m increases.
- The value of m where NMSE stabilizes and ceases to decrease significantly.
- The presence of any asymptotic behavior indicating the limit of accuracy for the given setup.

Minimum $\frac{m}{d}$ Required for Reliable Recovery

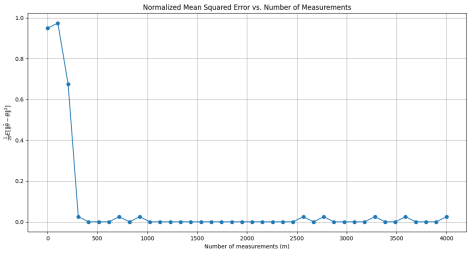
The minimum ratio of $\frac{m}{d}$ required for reliable recovery of θ is observed at the point where NMSE falls below a predetermined threshold. This threshold is application-dependent but we find here, looking at the plots to be ≤ 0.1.



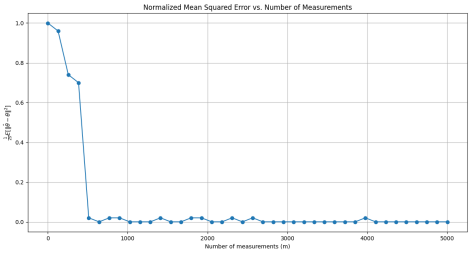
(a) $d = 2000$



(b) $d = 3000$



(c) $d = 4000$



(d) $d = 5000$

Figure 3: Plots of $\frac{1}{2s}E\|\hat{\theta} - \theta\|^2$ as a function of m for various dimensions d. In each case, β was set to 1.

Conclusion

From the plots, we conclude that the modified base chain allows the MH algorithm to recover θ with a number of measurements m on the order of $O(s\log(d))$. This is evidenced by the rapid decrease in NMSE with increasing m and its subsequent stabilization indicating successful signal recovery.

Question 3: Recovering sparse signal from 1-bit measurements

3.1

For 1-bit ± 1 measurements :

$$\mathbb{P}(y_i = 1|X,\theta) = \mathbb{P}((X\theta)_i + \xi_i \geq 0) = \frac{1}{2}\text{erfc}(-\frac{(X\theta)_i}{\sqrt{2}\sigma}), \text{ for } i = 1,...,\text{d}$$

$$\mathbb{P}(y_i = -1|X,\theta) = 1 - \mathbb{P}(y_i = 1|X,\theta), \text{ for } i = 1,...,\text{d}$$

And the corresponding Gibbs measure is:

$$\pi_\beta(\theta; y, X) = \frac{\prod_{i=1}^{m} e^{\beta \log(\mathbb{P}(y_i|X,\theta))}}{Z_\beta(y, X)}$$

Where $Z_\beta(y, X)$ is the normalization constant and is a sum over $\binom{d}{s}$ terms. For large d, computing π_β would still be very inefficient as computing $Z_\beta(y, X)$ is roughly of order $\mathcal{O}(md^{s+1})$ for $s \ll d$ (requires computing d^s matrix-vector multiplications).

3.2

Using the base chain defined in part 2.2 and simulated annealing, the following results were obtained for 1-bit measurements:

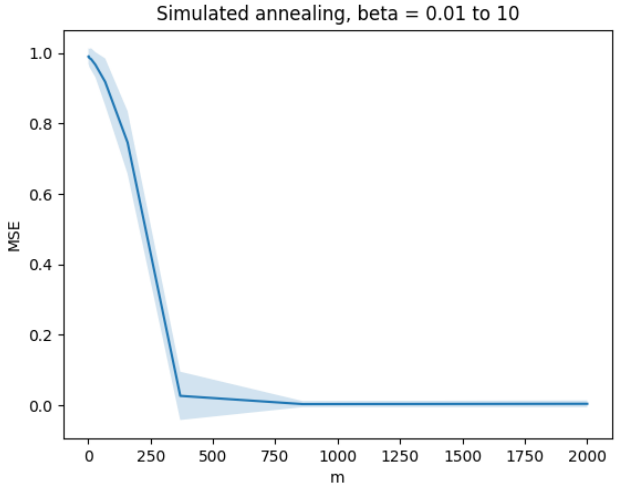


Figure 4: Same setup as figure 2 but with a sparsity s = d/100 and 1-bit measurements