MCMC Metropolis-Hastings: Ising Model for Image Denoising

Jiaming Li

Courant Institute, New York University

May 14, 2024

1 Introduction

Noise in images can arise due to various factors, including equipment limitations, poor lighting conditions, or errors during transmission, all of which degrade the quality of images. Effective techniques for denoising images are wanted and MCMC (Markov Chain Monte Carlo) methods are well-suited for such tasks since the nature of noise is usually a random distributed variable added to the original image. The Ising model — a statistical model originally developed in the field of statistical mechanics. Under the context of image processing, the Ising model treats image pixels like atomic spins that can either be in an "up" (+1) or "down" (-1) state, which correspond to different pixel colors. The model assumes that pixels prefer to be in the same state as their neighbors, which reflects the real-world situation where within small areas of an image, pixels tend to have similar colors. This report experiments with the Ising model within the framework of the Metropolis-Hastings algorithm to denoise binary images.

2 Mathematical Background and Scientific Notation

2.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is designed to generate a sequence of sample values from a probability distribution from which direct sampling is challenging. It works by creating a Markov chain that has the target distribution as its equilibrium distribution. The algorithm accepts or rejects randomly proposed moves (samples) based on how probable the move is under the target distribution, using an accept rate that ensures the detailed balance condition is met, thus converging to the target distribution.

1. Initialization: Let $f(x)$ be a function proportional to the target probability density function $P(x)$. Start with an arbitrary initial point x and choose a symmetric proposal function $g(x|y)$.
2. Iteration:
   - Propose a new candidate x' from $g(x'|x)$.
   - Calculate the acceptance ratio $\alpha = \dfrac{f(x')g(x_t|x')}{f(x_t)g(x'|x_t)} = \dfrac{f(x')}{f(x_t)} = \dfrac{P(x')}{P(x_t)}$, since $g(x|y)$ is a symmetric distribution, thus, $g(x'|x_t) = g(x_t|x')$
   - The proposed move is accepted with a probability equal to the minimum of 1 and the acceptance ratio. If rejected, the next sample is the same as the current one.
3. Repeat the process for a large number of iterations to allow the Markov chain to explore the space and approximate the target distribution.

Here is a pseudocode for the algorithm:

Algorithm 1 Metropolis-Hastings Algorithm

Initialize the starting point $x^{(0)}$ and set $t = 0$
while the stopping criterion is not satisfied **do**
    Propose $x' \sim g(x' \mid x^{(t)})$, where q is the proposal distribution
    Compute acceptance ratio $\alpha = \min\left(1, \frac{f(x')}{f(x^{(t)})}\right)$
    Draw $u \sim \text{Uniform}(0, 1)$
    if $u \leq \alpha$ **then**
        Accept the proposal: $x^{(t+1)} = x'$
    else
        Reject the proposal: $x^{(t+1)} = x^{(t)}$
    end if
    $t = t + 1$
end while

2.2 The Ising Model

Suppose the original image has only 2 types of pixels, and a pixel can either be -1 or +1. For simplicity, let -1 be a white pixel and +1 be a black pixel. So we can treat this binary image as a series of random variables $\{X_1, X_2, ..., X_n\}$, where X_i represents the i_{th} pixel. The Ising model suggests that the probability distribution, given by the Boltzmann distribution with $\beta \geq 0$, of a pixel is:

$$P_\beta(X) = \frac{e^{-\beta H(X)}}{Z_\beta},$$

- Z_β : the partition function, a normalization factor ensuring that probabilities sum to one.
- $H(X) = -\sum_{\langle i,j \rangle} J_{ij} X_i X_j - \mu \sum_j h_j X_j$: the first sum is over adjacent pairs of pixels X_i, X_j, J_{ij} is the interaction (enforcing effect) between pixels X_i, X_j, there larger J is, the more similarity exists among neighbour pixels. h_j indicates the noise level of the pixel X_j. The larger h_j, the lower the noise level existing in X_j.

Therefore, combine with the Metropolis-Hastings algorithm, we have the following scheme:

At iteration t, the current image has pixels $\mathbf{X}^{(t)} = \{x_1^{(t)}, x_2^{(t)}, ..., x_n^{(t)}\}$. Draw a pixel x_i' using a symmetric proposal distribution g where $g(x_i'|x_i^{(t)}) = g(x_i^{(t)}|x_i')$.

Now we decide if we want to flip the sign of the pixel or not ($x' = -x_i^{(t)}$) with the accept rate α that is determined as follows:

$$\alpha = \frac{f_\beta(x_i')}{f_\beta(x_i^{(t)})} = \exp\left(-2\beta\left(J_{ij} \sum_{\langle i,j \rangle} x_i^{(t)} x_j + \mu h_i x_i^{(t)}\right)\right)$$

Here we assume the noise added to the original image is random and non-discriminating towards specific pixels or areas. Therefore, it is reasonable to assume the interaction J_{ij} between any two pixels pair is consistent and the noise level h_i of a single pixel is constant. So we can simplify this rate as:

$$\alpha = \exp\left(-\lambda \sum_{\langle i,j \rangle} x_i^{(t)} x_j - \gamma x_i^{(t)}\right)$$

λ represents the degree of interaction between neighboring pixels. The higher the value of λ, the more likely it is that adjacent pixels will share the same color. γ indicates the level of noise present in the image. A higher value of γ corresponds to a lower noise level in the noisy image.

An example accept rate α in one flip step can be explained using Fig. 1:

1. Assume we propose a new candidate pixel $x' = -x_j = -1$ from the proposal distribution and it is circled in yellow. And we circle all of its neighbors in red to calculate the accept rate later.
2. The accept rate $\alpha = \exp(-\lambda \times 1 \times (1 \times 5 + (-1) \times 3) - \gamma \times 1) = \exp(-2\lambda - \gamma)$
3. With probability α we accept the proposed x' and flip x_j from 1 to -1, and with $1 - \alpha$ probability we keep x_j as 1.

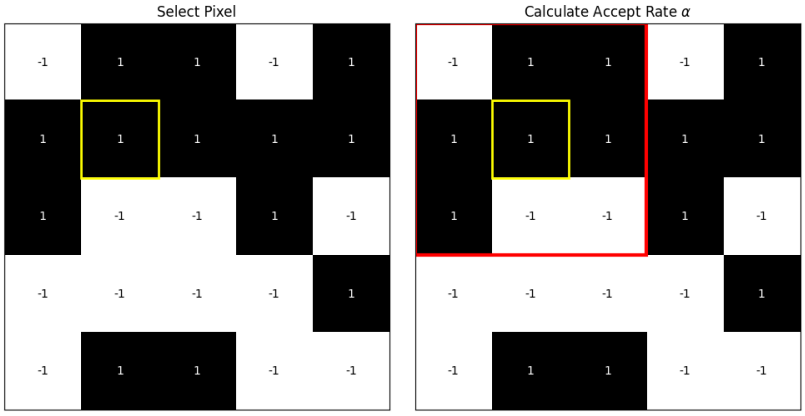

Fig. 1: Update step example

3 Numerical Tests and Results

Fig. 2: Original image and noisy image

Two images shown in Fig.2 are the original noise-free image (hidden) and noisy image (given). both of which are binary (black and white) and have a size of 640×640 pixels. In this experiment, we investigate two aspects of the algorithm: (1) tuning the parameters γ and λ, and (2) the proposal distribution function $g(x|y)$.

3.1 Parameter Tuning

Since γ and λ represent the noise level in the noisy image and the influence of neighboring pixels (indicating how strongly the algorithm adheres to the Ising model) respectively, the denoising algorithm is executed with different values of γ and λ. This allows us to explore the impact of these parameters on the denoised result. As shown in Fig. 3 and Fig. 4, a λ value of at least 1 is necessary for the algorithm to effectively denoise the image. When λ is too small, the influence of neighboring pixels in the Ising model is negligible.

This means that the pixel flips do not depend on the neighbors' colors, causing the algorithm to deviate from the Ising model, resulting in a failure to denoise the image. Instead, the image becomes increasingly noisy, making it impossible to recover any part of the original image (as seen in Fig. 4).

After 5×10^6 iterations, we can observe that for smaller γ values, the denoising process occurs faster but is less precise, resulting in the loss of finer details: the lip of Lena and shadows on her hat.

Fig. 3: 5×10^5 iterations

3.2 Proposal Distribution

The proposal distribution used in Fig. 3 and Fig. 4 is $g = U(1, 640)$. Regardless of the previous $x^{(t)}$, the next pixel is always drawn uniformly from the entire image. Now, we replace g with $g(x|y) = U(y-n, y+n)$, where n is a chosen number, and another alternative $g(x|y) = N(y, \sigma^2)$.

Fig. 4: 5×10^6 iterations

As shown in Fig. 5 and Fig. 6, a uniform distribution with a larger range and a Gaussian distribution with a larger σ produce better denoising results within the same number of iterations. Specifically, uneven patches and strips are observable in Fig. 5 when $n = 1$ and in Fig. 6 when $\sigma = 1$. This is likely due to the underlying nature of the noise, which is consistent throughout the entire image.

Fig. 5: Uniform distribution $g(x|y) = U(y - n, y + n)$, 5×10^5 iterations

Fig. 6: Gaussian $g(x|y) = N(y, \sigma^2)$, 5×10^5 iterations

Conclusions

In this report, we examined the application of the Ising model and Metropolis-Hastings algorithm for image denoising, focusing on the effects of parameter tuning and proposal distribution selection. The results demonstrated that a λ value of at least 1 is necessary for effective denoising, as smaller λ values result in negligible influence from neighboring pixels, causing the algorithm to fail in adhering to the Ising model. Additionally, it was observed that smaller γ values speed up the denoising process but at the cost of finer detail preservation. We also found that a uniform distribution with a larger range and a Gaussian distribution with a larger σ produced better denoising results within the same number of iterations. Uneven patches and strips were noted in cases of small n and σ, likely due to the consistent nature of the added noise. These findings underscore the importance of careful parameter selection and distribution function choice in optimizing the denoising performance of the algorithm.

References

1 Wong, Alexander, et al. "Stochastic image denoising based on Markov-chain Monte Carlo sampling." Signal Processing 91.8 (2011): 2112-2120.

2 GitHub repository: https://github.com/suyunu/Markov-Chain-Monte-Carlo.git