**Denoising of two levels by Metropolis-Hastings and Gibbs sampling**

**Introduction**

The purpose of this project is to denoise a corrupted image by estimating intensities in the original image through Metropolis-Hastings (MH) and Gibbs sampling. To simplify the problem, the original image has only two gray levels and is corrupted with Gaussian noise with known variance. The original image was generated by thresholding of the *House* image that is often used for digital image processing research. The *House* image and the modified image that will be referred to as the *original* image for the rest of this document is shown in Figure 1.

C:\Users\nnf001\Desktop\SSP\FinalProject\house.tif C:\Users\nnf001\Desktop\SSP\FinalProject\house_original.tif

*Figure 1. House image (left) and original image used for corruption and denoising (right)*

The original image was corrupted with Gaussian noise () to generate the *corrupted* image shown in Figure 2.

C:\Users\nnf001\Desktop\SSP\FinalProject\house_noisy.tif

*Figure 2. House image (left) and original image used for corruption and denoising (right)*

**Data Model**

The data model assumed for a single pixel *i* in corrupted image is that of a Gaussian mixture

Equation 1

where and are the original intensity values (80 and 170), and are both the standard deviation of the noise () added to the image and is the portion of pixels in the original image with gray level . Given the model for a pixel given in Equation 1, the model for a batch of pixels randomly chosen from the image is

Equation 2

where M is the number of pixels.

In this project three parameters will be estimated: , , and . The parameters and are assumed to be known. Table 1 shows the true values for the parameters and their classification as known or unknown.

*Table 1. List of parameters for the image data model*

|  |  |  |
| --- | --- | --- |
| Parameter | True value | Classification |
|  | 80 | Unknown |
|  | 170 | Unknown |
|  | .46 | Unknown |
|  | 30 | Known |
|  | 30 | Known |

**Posterior Density Function**

Bayes rule was used to estimate the posterior density function up to a normalizing constant

Equation 3

where represents the parameters , , and to be estimated. The likelihood is already given in Equation 2 but the prior was chosen after making some assumptions. It was assumed that there is no correlation between the parameters , , and their distributions were Gaussian. The distribution for was chosen to be *U(0,1)*. Additionally, there is a constraint such that . Under those circumstances, the prior is a multivariate Gaussian or zero,

Equation 4

Since this is a bivariate normal distribution, , is a 2x2 covariance diagonal matrix with variances along the diagonal (since there is no correlation) and is a vector with only two variables, and . In terms of and , Equation 4 is expressed as

Equation 5

where and are the variances for the distributions of and , respectively.

Finally, the posterior is given by Equation 3 as the product of the prior and the likelihood

Equation 6

**Metropolis-Hastings Algorithm**

An MH algorithm was used to sample the distribution of the parameters , , and . The pdf that was simulated was the posterior probability in Equation 6. The parameters for the simulation are shown in Table 2.

*Table 2. Parameters for the MH algorithm*

|  |  |
| --- | --- |
| Parameter | Value |
|  | 30 |
|  | 30 |
|  | 110 |
|  | 140 |
|  | 20 |
|  | 20 |
|  | .10 |

The value for the parameters and was assumed to be known, as mentioned in the Data Model section. The rest of the parameters, the prior parameters in Table 2 were chosen not too close to their true value because the purpose of the algorithm is to calculate their true values. Also, the jumping pdfs for , , and were parameterized by and , and , and , respectively.

At every iteration *i*, samples , , and were drawn from the jumping pdf. The acceptance probability for a combination of the three samples is given by

Equation 7

where is the target distribution. When , the new sample is updated with

= Equation 8

Otherwise, a sample is drawn from *U(0,1)* and compared against the acceptance probability to determine whether the sample will be taken or not.

**Gibbs Sampling Algorithm**

The Gibbs sampling algorithm is a special case of the MH algorithm type. The only difference is that the number of samples tested for acceptance is the same as the number of parameters to be estimated. In this particular implementation, in each iteration *i,* the samples for , , and are tested for acceptance separately. A sample of is first calculated and tested for approval. If it is accepted, then its value will be used in testing the approval of and . If the sample is approved, then its value will be used for testing of the sample .

**Simulation Results for MH**

A total of 100000 samples for each parameter were generated to test for acceptance by the MH algorithm. The accepted samples for , , and by the MH algorithm are shown in Figure 3.

C:\Users\nnf001\Desktop\SSP\FinalProject\mh_m1.tif C:\Users\nnf001\Desktop\SSP\FinalProject\mh_m2.tif C:\Users\nnf001\Desktop\SSP\FinalProject\mh_lambda.tif

*Figure 3. Accepted samples for , , and (from left to right)*

The values of the last samples were , , and

The last samples were used to denoise the corrupted image in Figure 2. This was done by first passing the image through a 3X3 median filter. The filter image was then thresholded with a threshold given by

Equation 9

Low values in the thresholded image were then assigned the value of and high values were assigned the value of . The resulting image, along with the original and corrupt images for comparison is shown in Figure 4.

C:\Users\nnf001\Desktop\SSP\FinalProject\house_original.tifC:\Users\nnf001\Desktop\SSP\FinalProject\house_noisy.tifC:\Users\nnf001\Desktop\SSP\FinalProject\house_denoised.tif

*Figure 4. Original, corrupt, and MH algorithm denoised image (from left to right)*

**Simulation Results for Gibbs Sampling**

For Gibbs sampling, only 1000 samples for each parameter were tested for acceptance. The resulting signals are shown in Figure 5.

C:\Users\nnf001\Desktop\SSP\FinalProject\gibbs_m1.tif C:\Users\nnf001\Desktop\SSP\FinalProject\gibbs_m2.tif C:\Users\nnf001\Desktop\SSP\FinalProject\gibbs_lambda.tif

*Figure 5. Accepted samples for , , and (from left to right)*

The last values for the parameter in the Gibbs simulation were , , and The resulting image, along with original and denoised images is shown in Figure 6.

C:\Users\nnf001\Desktop\SSP\FinalProject\house_original.tifC:\Users\nnf001\Desktop\SSP\FinalProject\house_noisy.tifC:\Users\nnf001\Desktop\SSP\FinalProject\house_denoised_gibbs.tif

*Figure 4. Original, corrupt, and Gibbs algorithm denoised image (from left to right)*

**Implementation (MatLab code):**

(the implementation is attached to this document)