STA5107 Midterm Project 1: Bayesian Analysis of Noisy Images

Jaime Frade

March 5, 2009

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

1.1.1 Problem Statement

The objective of this assignment is that if given observed noisy images, the goal is to perform a Bayesian analysis of the data. We will assume a prior probability model and an observation model to obtain a posterior density, and will generate samples from the posterior

Chapter 2

Methodology

2.1 General Approach

Given an distorted image, by the model $D = I + W$, where $W \sim N(0, \sigma_2^2)$, the approach must obtain an estimate for the posterior density of the image, $I \in \mathbf{R}^{m \times n}$. The image of I, a matrix of R.V's forming a Markov Random field, is a model that has prior knowledge. Each pixel, $I_{j,k}$ is a conditional density that is only dependent on the values of its vertical and horizontal neighbors of the given pixel. Given that μ is the mean of the horizontal and vertical pixels of $I_{j,k}$, if is known that $I_{j,k} \sim N(\mu, \sigma_1^2)$

2.1.1 Bayesian Analysis and Methods

Using Bayesian methods, we can estimate the posterior density of I, by sampling from the posterior of each $I_{j,k}$, given $D_{j,k}$ by using the following Bayesian rules.

$$f(I|D) = f(D|I)\frac{f(I)}{f(y)} \tag{2.1}$$

where $f(y) = \int f(D)f(I)dy$.

As stated above, since the conditional density is dependent only on certian neighbor of values, can form the following

$$f(I_{j,k}|D_{j,k}, I_{j+1,k}, I_{j-1,k}, I_{j,k+1}, I_{j,k-1}) = f(D_{j,k}|I_{i,j})f(I_{j,k} \tag{2.2}$$

Since $I_{j,k} \sim N(\mu, \sigma_1^2)$ and $D_{j,k} \sim N(I_{j,k}, \sigma_2^2)$ then

$$f(I_{j,k}|D_{j,k}, I_{j+1,k}, I_{j-1,k}, I_{j,k+1}, I_{j,k-1}) \sim N(\mu, \sigma_1^2) \cdot N(I_{i,j}, \sigma_2^2) \tag{2.3}$$

2.1.2 Metropolis Hastings

Metropolis Hastings is used in to approximate sampling from complicated distributions. In general, the goal is to generate samples of a random variable distributed according to the density, say $f(x)$. Moreover, we assume that the conditional density, say $q(y|x)$ with the following properities

1. $\forall\, x$, sampling from $q(y|x)$ is possible

2. The support of q contains the support of $f(x)$

3. $q(y|x)$ is known and symmetric in x and y.

Given a function and a conditional density with the above properities, the M-H algorithm is the following

1. Choose an intial condition x_0 in support of $f(x)$
   Construct x_n using the following steps:

2. Generate $y \sim q(y|x_t)$

3. Update the state to x_{t+1} by using

$$x_{t+1} = \begin{cases} y & \text{probabilty } \rho(x_t, y) \\ x_t & \text{probabilty } 1 - \rho(x_t, y) \end{cases} \tag{2.4}$$

where $\rho(x, y) = \min\left(\frac{f(y)q(x|y)}{f(x)q(y|x)}, 1\right)$. Under certain conditions, ρ can be simplifed, as such the case when

1. In cases where the density is independent of the current state, $q(y|x) = q(y)$, then becomes an independent M-H. Therefore the function becomes

$$\rho(x, y) = \min\left(\frac{f(y)q(x)}{f(x)q(x)}, 1\right) \tag{2.5}$$

2. When $q(y|x)$ is symmetric in x and y, then the likelihood rate appears, because the function becomes

$$\rho(x, y) = \min\left(\frac{f(y)}{f(x)}, 1\right) \tag{2.6}$$

2.1.3 Gibbs Sampler

Another technique for generating Markov Chains is the process of Gibbs Sampling. The goal is to generate samples by constructing a MC in $\mathbf{R}^n$, from a random vector, $(x_1, x_2, \ldots, x_n)$, with joint pdf, $f(x_1, x_2, \ldots, x_n)$. In order to use Gibbs sampling for this problem, will assume the conditional densities are known, so $f(x_i|y_i)$ for $i \neq j$. Therefore will obtain univariate densities to apply the algorithm to update from x^t to x^{t+1}

1. Generate $X_1^{t+1} \sim f_1(x_1|X_2^t, X_3^t, X_4^t)$

2. Generate $X_2^{t+1} \sim f_2(x_2|X_2^t, X_3^t, X_4^t)$

3. Generate x_1^{t+1}, x_2^{t+1}

In the procedure, each pixel, $I_{j,k}$ is processed until a complete sweep which will result in a new prior distrubtion, which in turn will be used in the next iteration. Therefore, in order to do a complete sweep, will sample from the posterior using the Gibbs sampling method, and update the posterior on each squenece.

Chapter 3

Matlab Code

3.1 Main Code

```
clear
clc

load DataFile1.mat
I=D1;
[n1,n2]=size(D1);
sigma1=10;
sigma2=30;

figure(1)
imagesc(I(:,:));
title('Initial Image');
saveas(figure(1),['Initial Image DataFile5.png']);

for i=1:6;
     for j=1:n1;
        for k=1:n2;
            mid = mean1(j,k,I);
            if rand>=0.5;
                I(j,k) = random('normal',mid,sigma1,1,1);
            else
                I(j,k)=I(j,k);
            end;
        end;
    end;

    W=random('normal',0,sigma2,n1,n2);
    D=I+W;

I2 = Gibbs(I,D,sigma1,sigma2);
I=I2;
```

```
figure(2)
subplot(2,3,i);
imagesc(I);
figname = sprintf('Image of sweep  %d',i+1);
title (figname);
saveas(figure(2),['Pictures of sweep' int2str(i) ' of DataFile1.mat (sigma=10).png']);
end;
```

3.2 MH Code

```
function [mid] = mean1(j,k,x)
          [n1,n2]=size(x);
            if (j==1) && (k==1)
               mid=(x(j,k+1)+x(j+1,k))/2;
            end;

            if (j==1) && (k==n2)
               mid=(x(1,k-1)+x(j+1,k))/2;
            end;

            if (j==n1) && (k==1)
               mid=(x(j-1,k)+x(j,k+1))/2;
            end;

            if (j==n1) && (k==n2)
               mid=(x(j,k-1)+x(j-1,k))/2;
            end;

            if (j==1 && k~=1 && k~=n2)
               mid=(x(j+1,k)+x(j,k-1)+x(j,k+1))/3;
            end;

            if (j==n1 && k~=1 && k~=n2)
               mid=(x(j-1,k)+x(j,k-1)+x(j,k+1))/3;
            end;

            if (j~=1 && j~=n1 && k==1)
               mid=(x(j-1,k)+x(j+1,k)+x(j,k+1))/3;
            end;

            if (j~=1 && j~=n1 && k==n2)
               mid=(x(j-1,k)+x(j+1,k)+x(j,k-1))/3;
            end;

            if (j~=1 && j~=n1) && (k~=1 && k~=n2)
               mid=(x(j-1,k)+x(j+1,k)+x(j,k+1)+x(j,k+1))/4;
            end;
```

3.3 Gibbs Code

```
function [gib] = Gibbs(I,D,sigma1,sigma2)
[n1,n2]=size(I);
    for j=1:n1;
        for k=1:n2;
  mid=mean1(j,k,I);
        mu = (mid/sigma1+D(j,k)/sigma2)*(1/((1/sigma1)^2 +(1/sigma2)^2));
        sd = sqrt(1/((1/sigma1)^2+(1/sigma2)^2));
        gib(j,k)=random('normal',mu,sd,1,1);
        end
    end
```

Chapter 4

Results

4.1 Comparing images for different σ_1

4.2 Plots Datafile1.mat

4.2.1 Orginal Plot of Datafile1.mat

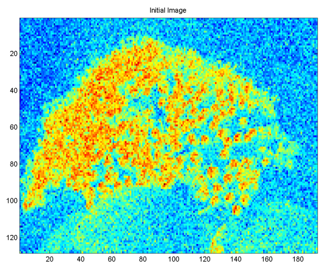
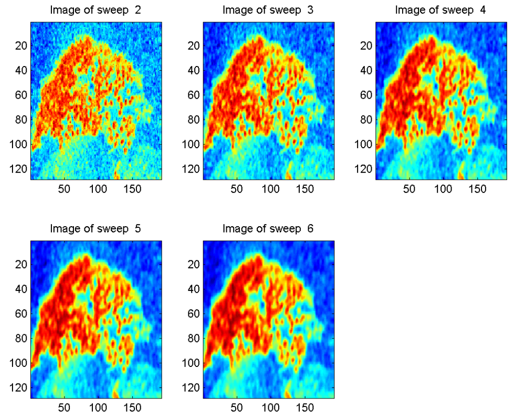


Image Datafile1.png

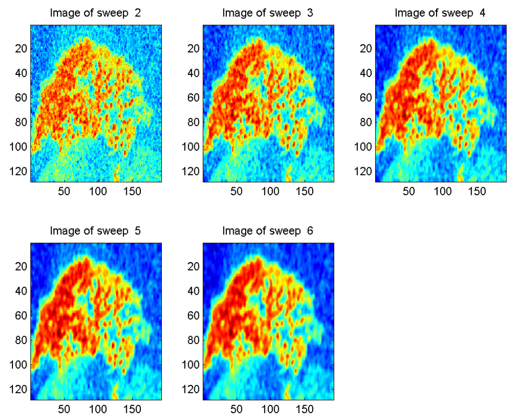
Figure 4.1: Orginal Image of Datafile1.mat

Images of Datafile1 after each sweep



of sweeps of datafile1.png

Figure 4.2: Image at each sweep for $\sigma_1 = 10$



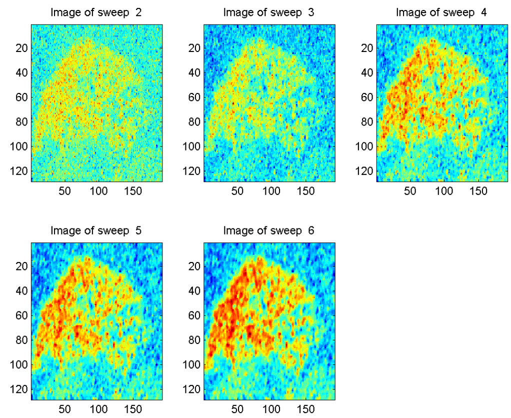
of sweep5 of datafile1 (sigma=20).png

Figure 4.3: Image at each sweep for $\sigma_1 = 20$

4.3 Plots Datafile2.mat

4.3.1 Orginal Plot of Datafile2.mat

Images of Datafile2 after each sweep



of sweep5 of datafile1 (sigma=100).png

Figure 4.4: Image at each sweep for $\sigma_1 = 100$

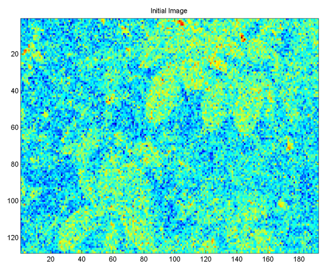


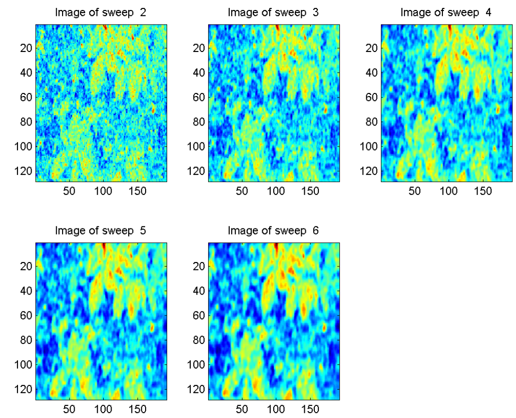
Image Datafile2.png

Figure 4.5: Orginal Image of Datafile2.mat

4.4 Plots Datafile3.mat

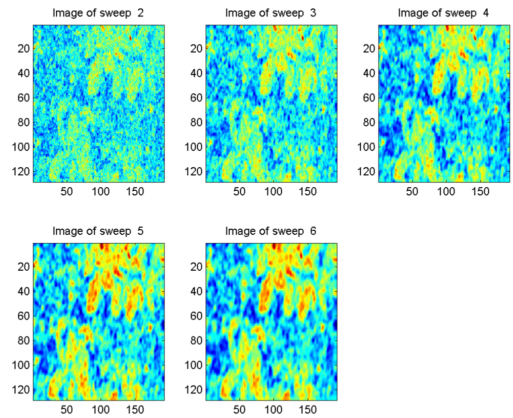
4.4.1 Orginal Plot of Datafile3.mat

Images of Datafile3 after each sweep



of sweep5 of DataFile2 (sigma=10).png

Figure 4.6: Image at each sweep for $\sigma_1 = 10$



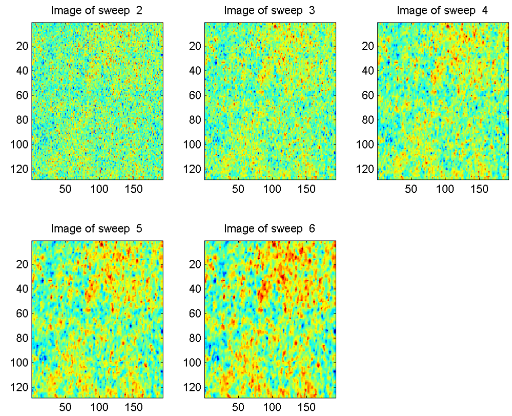
of sweep5 of Datafile2 (sigma=20).png

Figure 4.7: Image at each sweep for $\sigma_1 = 20$

4.5 Plots Datafile4.mat

4.5.1 Orginal Plot of Datafile4.mat

Images of Datafile4 after each sweep



of sweep5 of Datafile2 (sigma=100).png

Figure 4.8: Image at each sweep for $\sigma_1 = 100$

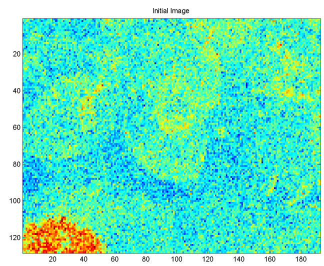


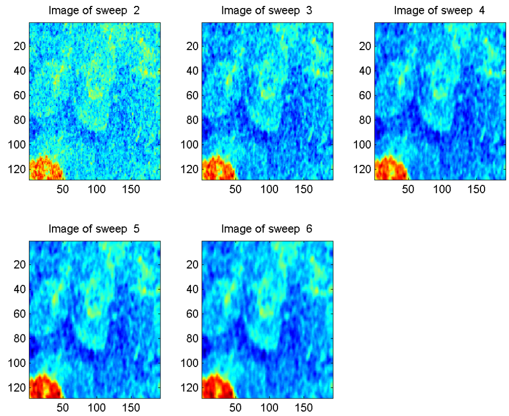
Image Datafile3.png

Figure 4.9: Orginal Image of Datafile3.mat

4.6 Plots Datafile5.mat

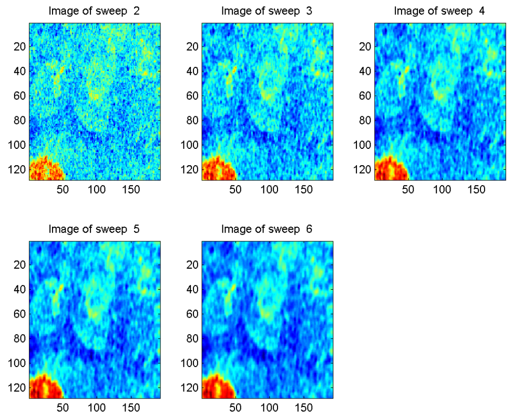
4.6.1 Orginal Plot of Datafile5.mat

Images of Datafile5 after each sweep



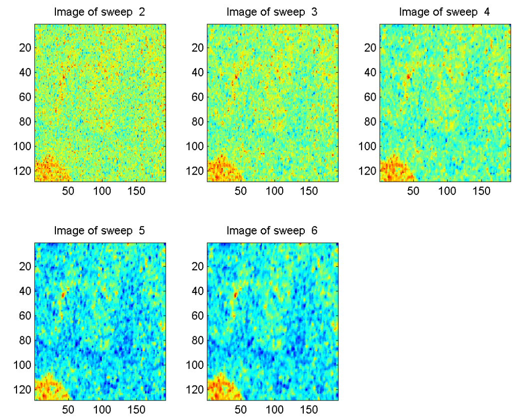
of sweep5 of Datafile3 (sigma=10).png

Figure 4.10: Image at each sweep for $\sigma_1 = 10$



of sweep5 of Datafile3 (sigma=20).png

Figure 4.11: Image at each sweep for $\sigma_1 = 20$



of sweep5 of Datafile3 (sigma=100).png

Figure 4.12: Image at each sweep for $\sigma_1 = 100$

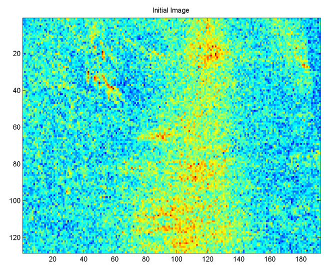
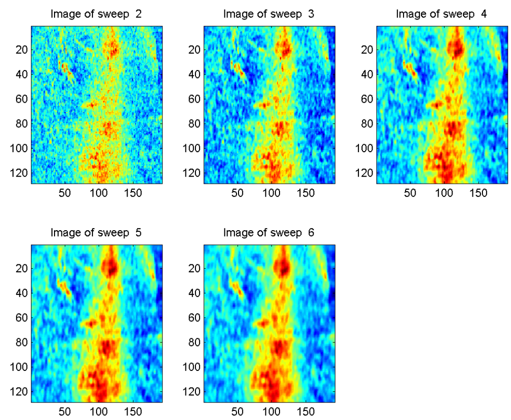


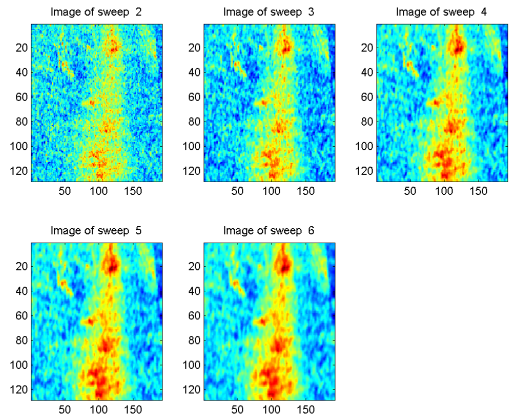
Image Datafile4.png

Figure 4.13: Orginal Image of Datafile4.mat



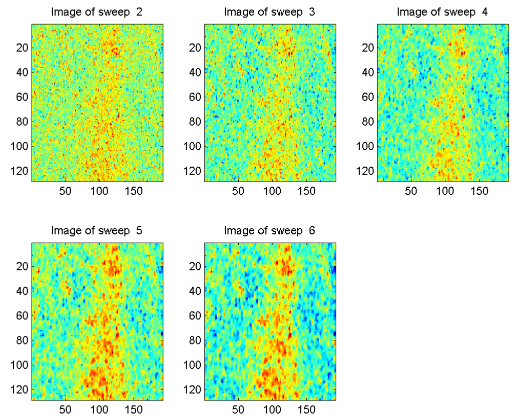
of sweep5 of Datafile4 (sigma=10).png

Figure 4.14: Image at each sweep for $\sigma_1 = 10$



of sweep5 of Datafile4 (sigma=20).png

Figure 4.15: Image at each sweep for $\sigma_1 = 20$



of sweep5 of Datafile4 (sigma=100).png

Figure 4.16: Image at each sweep for $\sigma_1 = 100$

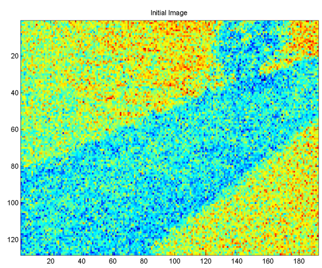
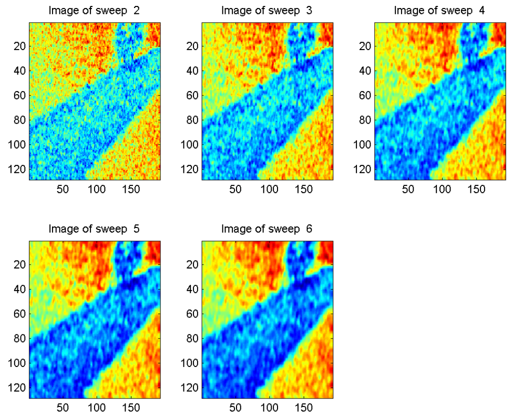


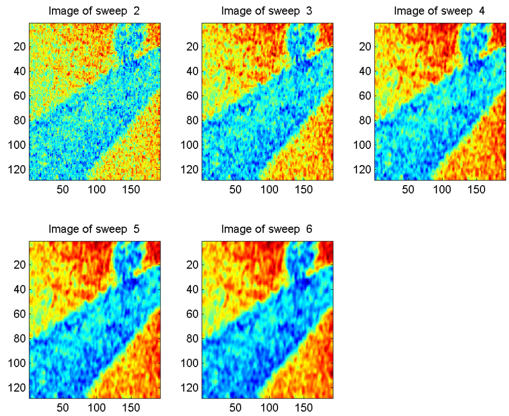
Image Datafile5.png

Figure 4.17: Orginal Image of Datafile5.mat



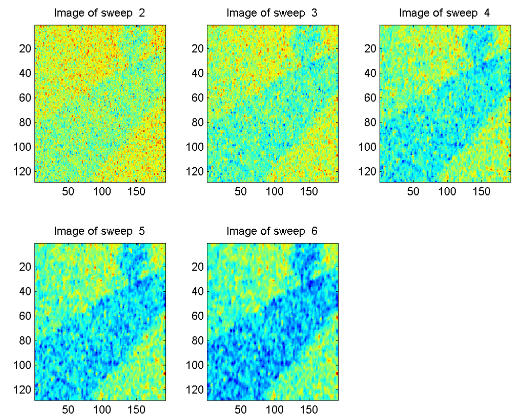
of sweep5 of Datafile5 (sigma=10).png

Figure 4.18: Image at each sweep for $\sigma_1 = 10$



of sweep5 of Datafile5 (sigma=20).png

Figure 4.19: Image at each sweep for $\sigma_1 = 20$



of sweep5 of Datafile5 (sigma=100).png

Figure 4.20: Image at each sweep for $\sigma_1 = 100$

Chapter 5

Conclusion

In this paper, it was found that the combinations of techniques applied above improved the qualty of the images at $\sigma_1 = 10$. As σ_1 increased, the images appeared to be more distorted, by adding more noise. As seen in plots, for smaller values of σ_1, the images are closer to I, where increasing σ_1, images appear closer to D