

Posterior Sampling with Proximal MCMC

Louis Christie
lgc26@cam.ac.uk

March 15, 2021

1 Introduction

In many image settings we are interested in the inverse problem of finding $x \in \mathbb{X}$ when we see $y \sim N(Ax, V)$ for example. One way to do this is to find a posterior distribution

$$\pi_{X|Y}(x | y) \propto \pi_{Y|X}(y | x) \times \pi_X(x) = N(y | Ax, V) \times \pi_X(x) \quad (1.1)$$

This requires us to specify a prior $\pi_X(x)$, which is often done by training a neural network or other model on a set of accurate images.

Once we've specified the prior and likelihood, we can estimate x from the posterior. One way is to find the Maximum A Posteriori (MAP) estimate $\hat{x}^{MAP} = \arg \max_{u \in \mathbb{X}} \pi_{X|Y}(u | y)$. Another is to find the posterior expectation $\hat{x} = \mathbb{E}(X | y)$. Since it may not be possible to analytically find this expectation, we turn to Monte-Carlo integration:

$$\mathbb{E}(X | Y) \approx \frac{1}{m} \sum_{i=1}^m X_i \quad \text{where } X_i \stackrel{iid}{\sim} \pi_{X|Y} \quad (1.2)$$

To do this we need to sample from the posterior, which we do by designing a Markov Chain with the target posterior as its stationary distribution. This also has the benefit of allowing us to quantify the uncertainty in our estimate using the sample variance of these samples. In our case we can use the MYULA chain

Proposition 1.1. *If $-\log \pi_{X|Y} = f + g$ where f is convex, proper, lower semicontinuous and has L_f -Lipschitz gradient and g is convex, proper and lower semicontinuous, then the chain with transitions:*

$$X_{n+1} = X_n - \delta \nabla f(X_n) - \frac{\delta}{\lambda} (X_n - \text{Prox}_g^\lambda(X_n)) + \sqrt{2\delta} Z_{n+1} \quad (\text{MYULA})$$

(where $Z_i \stackrel{iid}{\sim} N(0, 1)$ and $\delta < 2\lambda(L_f\lambda + 1)^{-1}$) converges exponentially fast to $\pi_{X|Y}$ as $n \rightarrow \infty$.

From equation 1.1 we see that for some normalisation constant C :

$$-\log \pi_{X|Y} = -\log C - \log \pi_{Y|X} - \log \pi_X \quad (1.3)$$

So we can set $f = -\log C - \log \pi_{Y|X}$ and $g = -\log \pi_X$ to use MYULA. This requires computing $\text{Prox}_g^\lambda(X_n) = \arg \min_{u \in \mathbb{X}} (g(u) + (2\lambda)^{-1} \|x - u\|_{\mathbb{X}}^2)$ at each step. This is a strongly convex function so does have a unique minimiser, but we would like to compute it without a descent style algorithm. The question becomes can we learn a prior π_X such that this proximal operator can be computed analytically?

Since g is sub-differentiable, we can define see that the proximal operator can be rewritten as an inverse image:

$$\text{Prox}_g^\lambda(x) = (I_{\mathbb{X}} + \lambda \partial g)^{-1}(\{x\}) \quad (1.4)$$

This can be seen as if $u \in (I_{\mathbb{X}} + \lambda \partial g)(x)$ then $0 \in \lambda \partial g(u) + u - x$ so u is a minimiser of $g(u) + \|x - u\|_{\mathbb{X}}^2 / (2\lambda)$. Since this is strongly convex, the inverse image is a singleton.

[cite Subhadip](#) have shown that they can learn g as an **input convex neural network** (ICNN) by using nodes of the form:

$$z_{i+1} = \phi_i(B_i(z_i) + W_i(x_i) + b_i)$$

If the weights B_i are all non-negative and the ϕ_i are each convex and monotone then the learned network is also convex.

If we use a smooth activation function ϕ_i , this is also guaranteed to be smooth. This means that we can ignore the proximal step in MYULA entirely - we can just use the smooth posterior to sample easily.

2 Gaussian Example

Suppose that A is the identity operator, so $y \mid x \sim N(x, \Sigma_\epsilon)$. If we take a Gaussian prior on X , so $\pi_X(x) = N(x \mid 0, \Sigma_X)$, then we have a known posterior:

$$\pi_{X|Y}(x \mid y) \propto N(x \mid y, \Sigma_\epsilon) \times N(x \mid 0, \Sigma_X) \quad (2.1)$$

$$\propto N(x \mid S(\Sigma_\epsilon^{-1}y), S) \quad (2.2)$$

where $S^{-1} = \Sigma_\epsilon^{-1} + \Sigma_X^{-1}$.

Example 2.1. In Willem's example, we have $y = (1, 1)$, $\Sigma_\epsilon = \frac{1}{2}I_2$, and $\Sigma_X = I_2$. Thus we have the posterior

$$\pi_{X|Y}(x \mid (1, 1)^T) = N\left(x \mid \begin{pmatrix} \frac{3}{2} \\ \frac{3}{2} \end{pmatrix}, \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3} \end{pmatrix}\right) \quad (2.3)$$

Sampling 4000 steps of MYULA with 1000 to burn in gives a sample with a KDE given in figure 2.1.

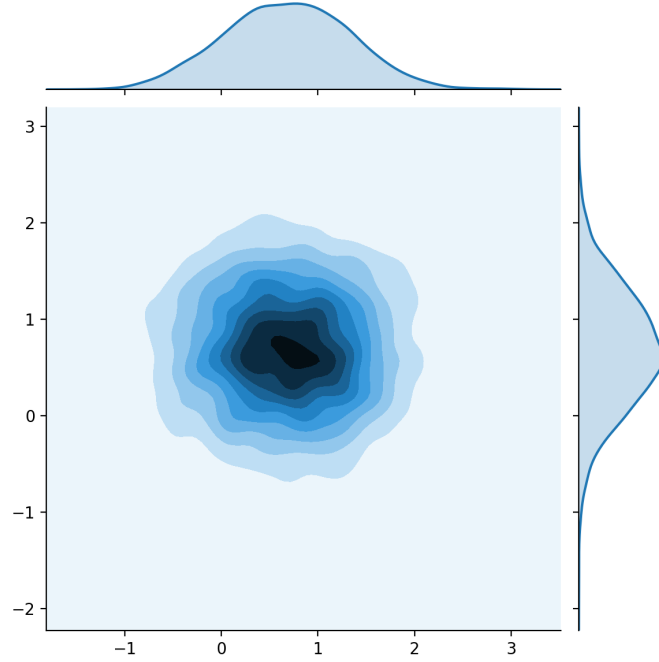


Figure 1: Posterior Sample KDE from MYULA. Sample mean at $(0.682, 0.681)^T$ with marginal variances of 0.42 and 0.38

3 Lasso Prior

In the same context as above we have the likelihood $N(y \mid Ax, \Sigma_\epsilon)$, but now we have $-\log \pi_X(x) = g(x) = \|x\|_1$. Thus the prox of g is given by:

$$\text{Prox}_g^\lambda(x) = \arg \min_{u \in \mathbb{R}^d} \frac{1}{2} \|u - x\|_2^2 + \lambda \|u\|_1 \quad (3.1)$$

As the objective is separable component wise; we get:

$$\text{Prox}_g^\lambda(x)_i = \arg \min_{u_i \in \mathbb{R}} \frac{1}{2} (u_i - x_i)^2 + \lambda |u_i| \quad (3.2)$$

$$= \{u_i \in \mathbb{R} : 0 \in \partial \frac{1}{2} (u_i - x_i)^2 + \lambda |u_i|\} \quad (3.3)$$

$$= \{u_i \in \mathbb{R} : 0 \in (u_i - x_i) + \lambda \text{sign}(u_i) \mathbf{1}_{u_i \neq 0} + \lambda \mathbf{1}_{u_i = 0} [-1, 1]\} \quad (3.4)$$

$$= \text{sign}(x_i) \max(0, |x_i| - \lambda) \quad (3.5)$$

This is called the **soft-thresholding operator**.

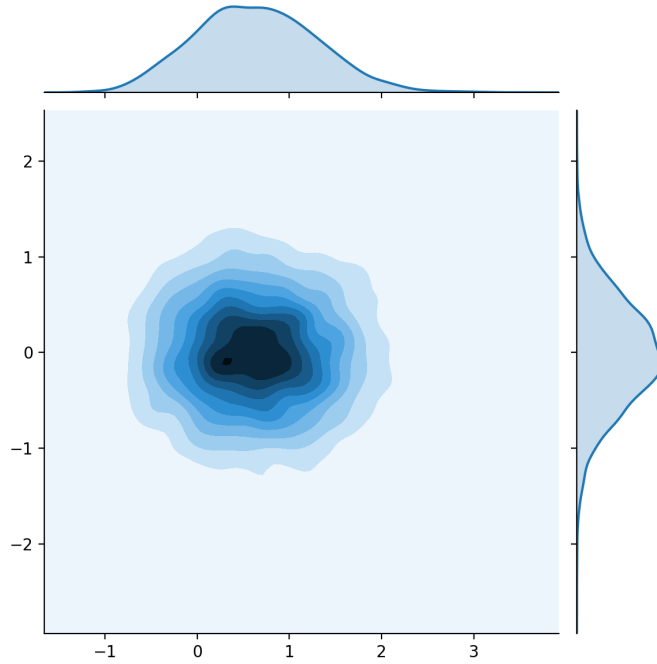


Figure 2: Posterior Sample KDE from MYULA. Sample mean at $(0.639, -0.019)^T$ with marginal variances of 0.44 and 0.34

4 Total Variation Prior

5 Wavelet Prior

We can transform the image into a wavelet domain and penalise the coefficients in this new basis.

6 Smooth Input Convex Neural Networks

The nodes of ICNNs are of the form:

$$z_{i+1} = \phi_i(B_i(z_i) + W_i(x_i) + b_i)$$

so if we use a smooth activation function $\phi_i = \text{softmax}$ we will have a smooth g , allowing us to ignore the prox step entirely.