Reproduction of image segmentation analysis using reversible jump Markov chain Monte Carlo

MATH 598 - Project 4

Larry Dong
Department of Epidemiology, Biostatistics and Occupational Health
McGill University
Montreal, Canada H3A 0G4
larry.dong@mail.mcgill.ca

May 4, 2019

1 Introduction (Problem Setup)

The image is partitioned into identically shaped pixels where its respective grayscale intensity is a number between 0 and 2. The image at hand consists of 50 by 50 pixels for a total of 2500 pixels. The target and original image of the black disk is available in Figure 1; it has a radius of 15 pixels. As detailed in Green (1995), the pixels within the circle has an intensity of 2.0 whereas the "background" pixels have an intensity of 0.5. In Figure 2, every pixel was degraded with a random Gaussian noise sampled from $\mathcal{N}\left(0, 0.7^2\right)$. However, for computational purposes, any grayscale intensity value that surpassed 2.0 was kept at 2.0 and any intensity that was negative was set to 0.

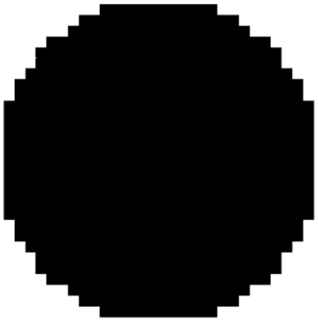

Figure 1: Target image or image before the addition of random Gaussian noise

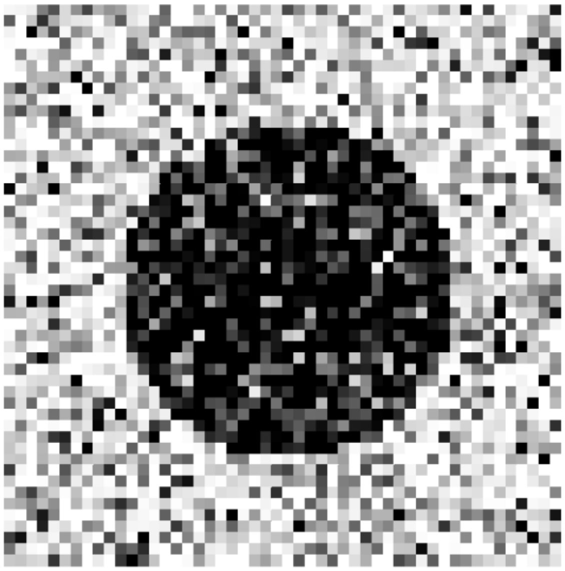

Figure 2: Noisy image

In subsequent statistical analyses and computational simulations, every pixel was labelled with a unique index as showcased in Figure 3:

1	2	3	$\ldots$	$\ldots$	49	50
51	52	53	$\ldots$	$\ldots$	99	100
101	102	103	$\ldots$	$\ldots$	149	150
$\vdots$	$\vdots$	$\ddots$	$\ldots$	$\ldots$	$\iddots$	$\vdots$
2451	2452	2453	$\ldots$	$\ldots$	2499	2500

Figure 3: Labelling of each pixel with unique indices

For instance, when refering to any pixel i for $i \in \{1, \ldots, 2500\}$, the horizontal midpoint u_i and vertical midpoint v_i are chosen to represent the pixel. When computing the distance between some other point $(u', v') \in [0, 50] \times [0, 50]$, the distance between a particular pixel i and (u', v') would be the $l2$-norm between (u_i, v_i) and (u', v').

2 Methods

2.1 Derivations

The parameters of interest in this model are $\{k\} \cup \theta^{(k)}$ where $\theta^{(k)} = \{(u_i, v_i, h_i)\}_{i=1}^{k}$ is the $3k$-dimensional vector of parameters who dimension depends on k. Let $\mathbf{y} = \{y_j\}_{j=1}^{2500} = \{y(u_j, v_j)\}_{j=1}^{2500}$ be the collection of observed pixel grayscale intensities as defined above, i.e. $y_i \in [0, 2]$ due to truncation of grayscale intensities as described in section 2.

The following distribution were posited for the data at hand and the parameters of interest:

$$
\begin{align}
Y_i \mid k, \theta^{(k)} &\sim \exp\left\{ -\frac{y(u_i, v_i) - x(u_i, v_i)}{2\sigma^2} \right\} \tag{1}\\
u_i &\sim \mathcal{U}(0, 50) \qquad \text{for } i \in \{1, \ldots, k\} \tag{2}\\
v_i &\sim \mathcal{U}(0, 50) \qquad \text{for } i \in \{1, \ldots, k\} \tag{3}\\
h_i &\sim \text{Gamma}(\alpha, \beta) \qquad \text{for } i \in \{1, \ldots, k\} \tag{4}\\
k &\sim \text{Poisson}(\lambda) \tag{5}
\end{align}
$$

where $y_i = y(u_i, v_i)$ and the constants α, β, λ and σ were chosen to be $1, 1, 15$ and 0.7 respectively. Note that an upper bound value for k was imposed to be $k_{\max} = 30$; however, the sampled values for k were nowhere near this upper bound (c.f. section 4). With this, the posterior distribution $\pi_n(k, \theta^{(k)}) = p(k, \theta^{(k)} \mid \mathbf{y})$ can be derived as followed:

$$
\begin{align}
\pi_n\left(k, \theta^{(k)}\right) &\propto \underbrace{L\left(\mathbf{y} \,\middle|\, k, \theta^{(k)}\right)}_{\text{Likelihood}} \cdot \underbrace{\pi_0(k) \prod_{i=1}^{k} \pi_0(u_i, v_i, h_i)}_{\text{Prior}} \tag{6}\\
&\propto \exp\left\{ -\frac{1}{2\sigma^2} \sum_{j=1}^{2500} \left(y(u_j, v_j) - x(u_j, v_j)\right)^2 \right\} \pi_0(k) \prod_{i=1}^{k} \underbrace{\pi_0(u_i)\pi_0(v_i)}_{\text{constant}} \pi_0(h_i) \tag{7}\\
&\propto \exp\left\{ -\frac{1}{2\sigma^2} \sum_{j=1}^{2500} \left(y(u_j, v_j) - x(u_j, v_j)\right)^2 \right\} \pi_0(k) \prod_{i=1}^{k} h_i^{\alpha-1} \exp\big(-\beta h_i\big) \tag{8}\\
&\propto \exp\left\{ -\frac{1}{2 \cdot 0.7^2} \sum_{j=1}^{2500} \left(y(u_j, v_j) - x(u_j, v_j)\right)^2 \right\} \pi_0(k) \prod_{i=1}^{k} \exp\big(-h_i\big) \tag{9}
\end{align}
$$

Notice $\pi_0(u_j) = \frac{1}{50}\mathbb{1}_{0<u_j<50}$ and $\pi_0(v_j) = \frac{1}{50}\mathbb{1}_{0<v_j<50}$ and these probability distributions can be dropped when evaluating the proportionality of the posterior distribution. Recall that $x(u_j, v_j) = h_i$ where $i =$

$\text{argmin}_{i' \in \{1,\dots,k\}} \left((u_j - u_{i'})^2 + (v_j - v_{i'})^2\right)$; in other words, the height assigned to pixel (u_j, v_j) is chosen to be the height assigned closest centroid. That being said, there is some set $H_i = \{j \,|\, x(u_j, v_j) = h_i\}$ that contains the indices of all points (u_j, v_j) where the assigned height $x(u_j, v_j) = h_i$. By the construction of the Voronoi diagram through the k sampled centroids which are all contained in the $[0, 50] \times [0, 50]$ 2D frame, it follows that $H_i \neq \emptyset$ for all $i \in \{1, \dots, k\}$, at any iteration of the Gibbs sampler algorithm. Most importantly, in terms on deriving the full conditional distribution, especially for h_i, the likelihood $L(\mathbf{y} \,\big|\, k, \theta^{(k)})$ in fact does depend on the latter:

$$
\begin{aligned}
\pi_n\left(u_i \,|\, k, \theta^{(k)} \backslash \{u_i\}\right) &\equiv \mathcal{U}(0, 50) \\
\pi_n\left(v_i \,|\, k, \theta^{(k)} \backslash \{v_i\}\right) &\equiv \mathcal{U}(0, 50) \\
\pi_n\left(h_i \,|\, k, \theta^{(k)} \backslash \{h_i\}\right) &= \exp\left\{-\frac{1}{2\sigma^2} \sum_{j \in H_i} \left(y(u_j, v_j) - h_i\right)^2\right\} \cdot \exp\left(-h_i\right) \\
&\overset{h_i}{\propto} \exp\left\{-\frac{1}{2\sigma^2}\left(h_i^2 n_i - 2h_i \sum_{j \in H_i} y(u_j, v_j) + \sum_{j \in H_i} y(u_j, v_j)^2 + 2\sigma^2 h_i\right)\right\} \\
&\propto \exp\left\{-\frac{1}{2\sigma^2}\left(\frac{h_i^2 - 2h_i\left(\sum_{j \in H_i} y(u_j, v_j) - \sigma^2\right)/n_i}{1/n_i}\right)\right\} \\
&\propto \exp\left\{-\frac{1}{2\left(\frac{\sigma}{\sqrt{n_i}}\right)^2}\left(h_i^2 - 2h_i\left(\frac{\sum_{j \in H_i} y(u_j, v_j) - \sigma^2}{n_i}\right) \pm \left(\frac{\sum_{j \in H_i} y(u_j, v_j) - \sigma^2}{2n_i}\right)^2\right)\right\} \\
&\propto \exp\left\{-\frac{1}{2\left(\frac{\sigma}{\sqrt{n_i}}\right)^2}\left(h_i - \left(\frac{\sum_{j \in H_i} y(u_j, v_j) - \sigma^2}{2n_i}\right)\right)^2\right\} \\
&\equiv \mathcal{N}\left(\frac{\sum_{j \in H_i} y(u_j, v_j) - \sigma^2}{2n_i}, \; \frac{\sigma^2}{n_i}\right)
\end{aligned}
$$

where $n_i = |H_i|$. Note that the full conditional posterior distributions of u_i and v_i for all $i \in \{1, \dots, k\}$ is uniform because the expression of the posterior distribution $\pi_n\left(k, \theta^{(k)}\right)$ do no depend on the locations of the Voronoi centroids. The sampling procedure for the number of Voronoi areas k follows a birth and death process whereby, in a given iteration of the Gibbs sampler, the probability of the addition of a new centroid is $\min(1, R)$ and the probability of removing an existing centroid is $\min\left(1, R^{-1}\right)$ where R is computed as per Green (1995) [5]. Note that, at every iteration, either a birth or a death process will proceed, and occasionally both processes will occur.

2.2 Sampling procedure $x(u_j, v_j)$

A Gibbs sampler method was implemented in order to sample from the posterior distribution [1, 4]. Since the parameter of interest is multidimensional, a Gibbs sampling procedure can be used, even when the dimension of the parameter space varies between iterations. In words, given k, the location of centroids $\{u_i, v_i\}_{i=1}^k$ is sampled from a uniform distribution and the heights h_i are sampled from a normal distribution as specified in the section above conditioned on the $\{u_i, v_i\}_{i=1}^k$. Then, the R statistic is computed and the algorithm then determines the new value of k depending if a birth or a death or both procedures will occur; in either of the three cases, the heights corresponding to the temporary Voronoi cells are updated accordingly and the values $x(u_j, v_j)$ are computed for all $j \in \{1, \dots, 2500\}$. The algorithm was run for $24\,000$ iterations where the sampled values from the first 4000 iterations were discarded as burn-in, as per Green (1995) [5]. The results of the simulation are displayed in the following section.

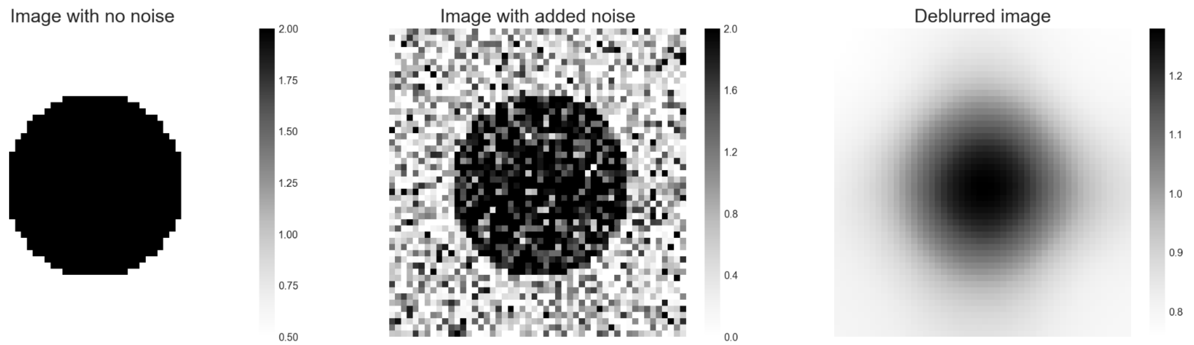

Figure 4: Comparison between the target image, the noisy image given as input to the model and the recovered image.

2.2.1 Sampling v

When considering a proposed birth of a generated point k^* whose location is (u_{k^*}, v_{k^*}), its newly assigned height is $h^* = v \cdot \left(\prod_{i \in \mathcal{J}} h_i^i\right)^{1/\sum_i i}$ where V is sampled from the following probability density function [5]:

$$f(v) = \frac{5v^4}{(1+v^5)^2}$$

Using the fact that random variable can be sampled if its continuous distribution function (CDF) is known, since any CDF a random variable is uniformly distribution on $(0, 1)$, it follows that $F^{-1}(U) \sim F_v$ where F_v is the CDF of V [1].

$$F_v(v) = \int_0^v f(t)dt = 1 - \frac{1}{1+v^5} \overset{\text{set}}{=} u$$
$$\Rightarrow v = \left(\frac{1}{1-u} - 1\right)^{\frac{1}{5}} = F_v^{-1}(u)$$

2.2.2 Computational restrictions

Two minor adaptations have been made to ensure the smooth runtime process of the suggested Gibbs sampling method described in the subsection above. As described above, at every iteration, the Voronoi centroids are sampled from the full conditional posterior distribution, which happens to be $\mathcal{U}(0, 50)$ as well. Every area of every Voronoi cell is then computed using the ConvexHull algorithm, which receives the coordinates of all vertices of the subregion and computes the area enclosed by them. It is important to note that this method

One other minor adaptation in the implementation of the sampling method described above is that the value k was chosen to have a lower bound of 3 rather than 1. Although the Voronoi tesselation is permissible with two or even one point, its implementation was difficult and we have opted to instead impose a restriction of $k \geq 3$. In other words, at a given time iteration where $k = 3$, then a death process is not possible.

Finally, all exploratory and inferential analyses were performed in Python version 3.6 using Numpy version 1.15.0, Scipy version 1.1.0, Seaborn version 0.9.0 and Matplotlib version 3.0.3.

3 Results and Discussion

The posterior mean of $x(u_j, v_j)$ estimated through the sampled values are plotted in Figure 4 alongside the target image and the noisy image which served as input to the Bayesian model.

Overall, the deblurred or recovered image seems to be ressemble the target image visually speaking. However, some flaws still persist in the obtained image. For instance, the pixels that surround the black disk are gray, i.e. the grayscale value of their intensities are, on average, higher compared to the pixels further from the disk. One other important flaw that can be pointed out is the scale of the intensities in the image. The noisy image has intensities varying from 0 to 2 inclusively, whereas the discrepancy between the grayscale intensity values of “white" and “black" pixels in the

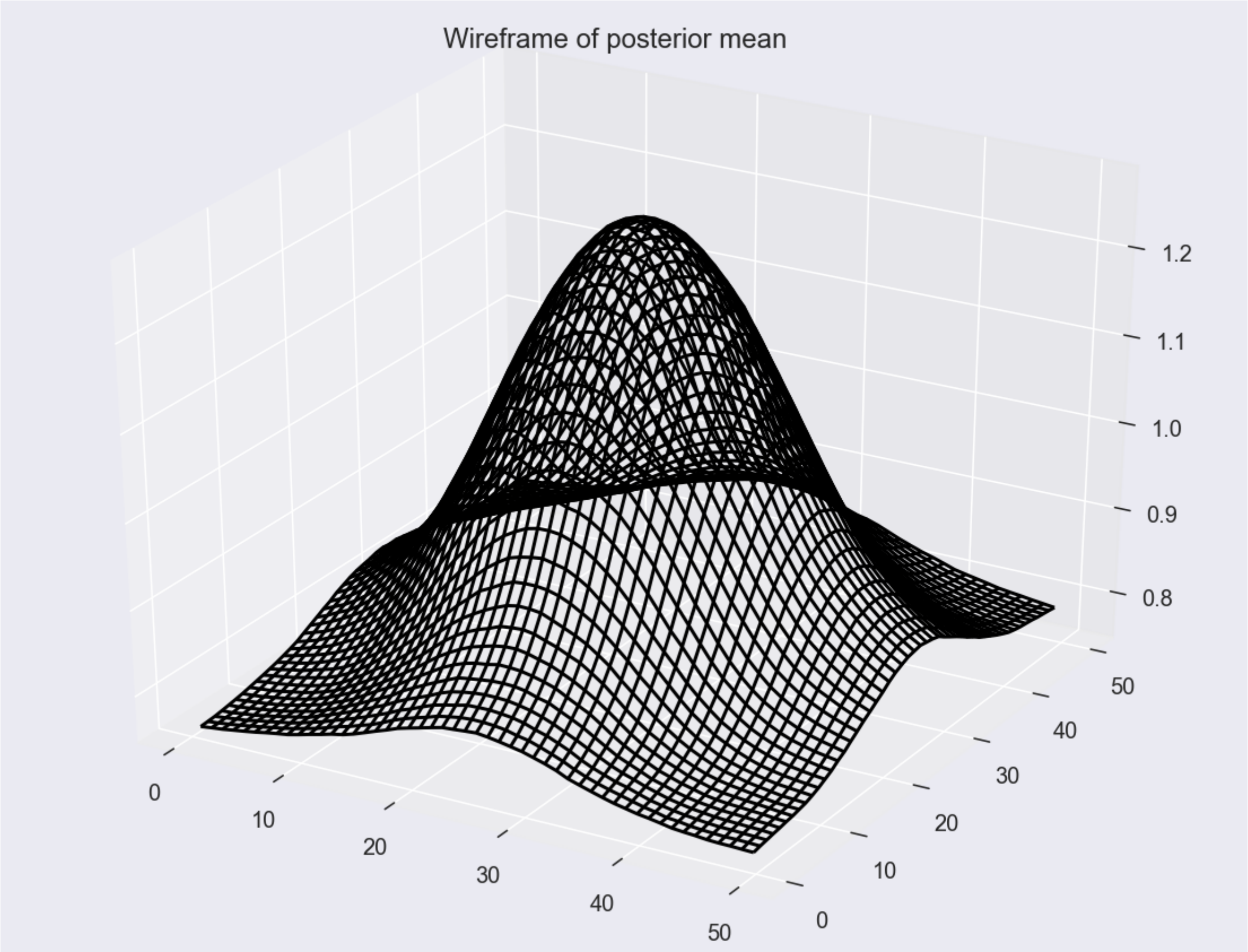

Figure 5: Wireframe of the estimated posterior mean of $x(u_j, v_j)$.

deblurred image is not as large.

While the Markov chain Monte Carlo (MCMC) procedure seems to converge (c.f. Figures 6 and 7), the wireframe in Figure 5 showcases a very “smooth" pattern among the sampled means of grayscale intensities. As said above, broadly speaking, the similarity between the deblurred image demonstrate that the implemented algorithms worked relatively well. That being said, important differences between the results obtained by Green (1995) and the ones showcased in Figure 4 may be explained through limitations addressed in the earlier sections of this report [5].

References

[1] Robert, C. P., & Casella, G. (2011). *Monte Carlo statistical methods*. New York: Springer.

[2] Brown, K. Q. (1979). Voronoi diagrams from convex hulls, *Information Processing Letters*

[3] Hastings, W. K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*.

[4] Gilks, W. R., Best, N. G., Tan, K. K. C. (1995). Adaptive Rejection Metropolis Sampling within Gibbs Sampling. *Journal of the Royal Statistical Society*. Series C (Applied Statistics).

[5] Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*.

Appendix A - Traceplots

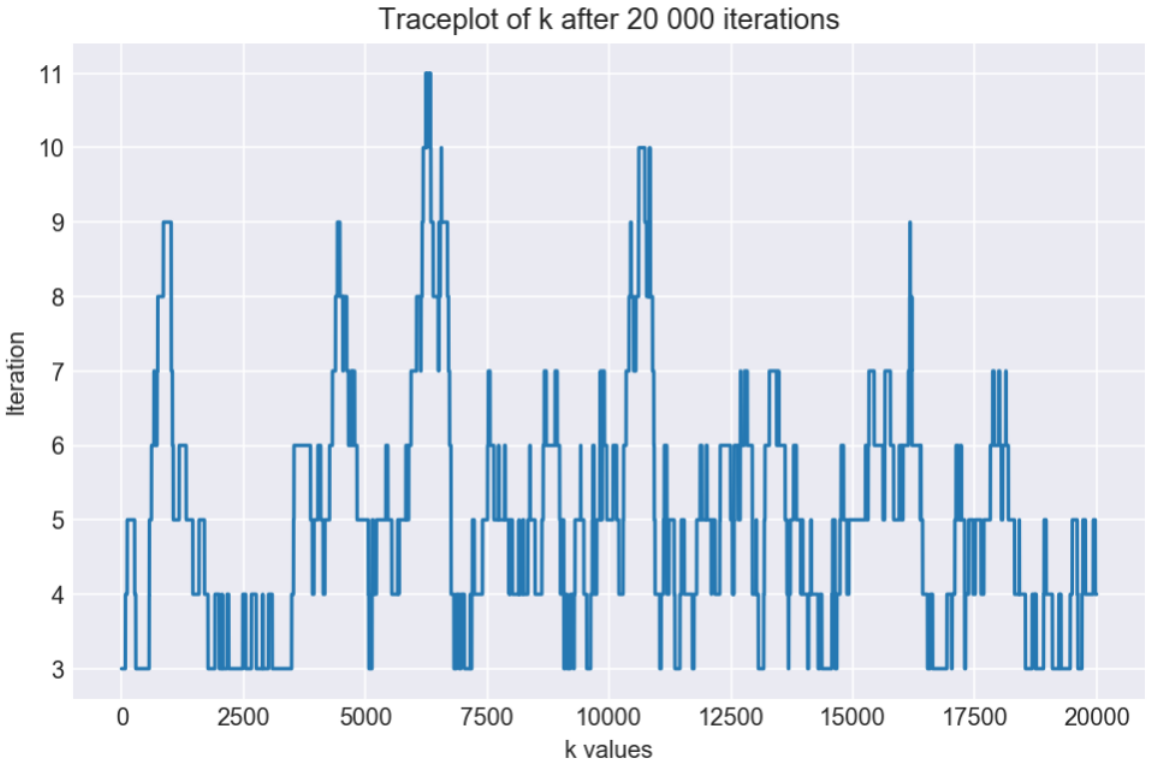

Figure 6: Traceplot for k after $20\,000$ iterations with a burn-in of 4000 iterations.

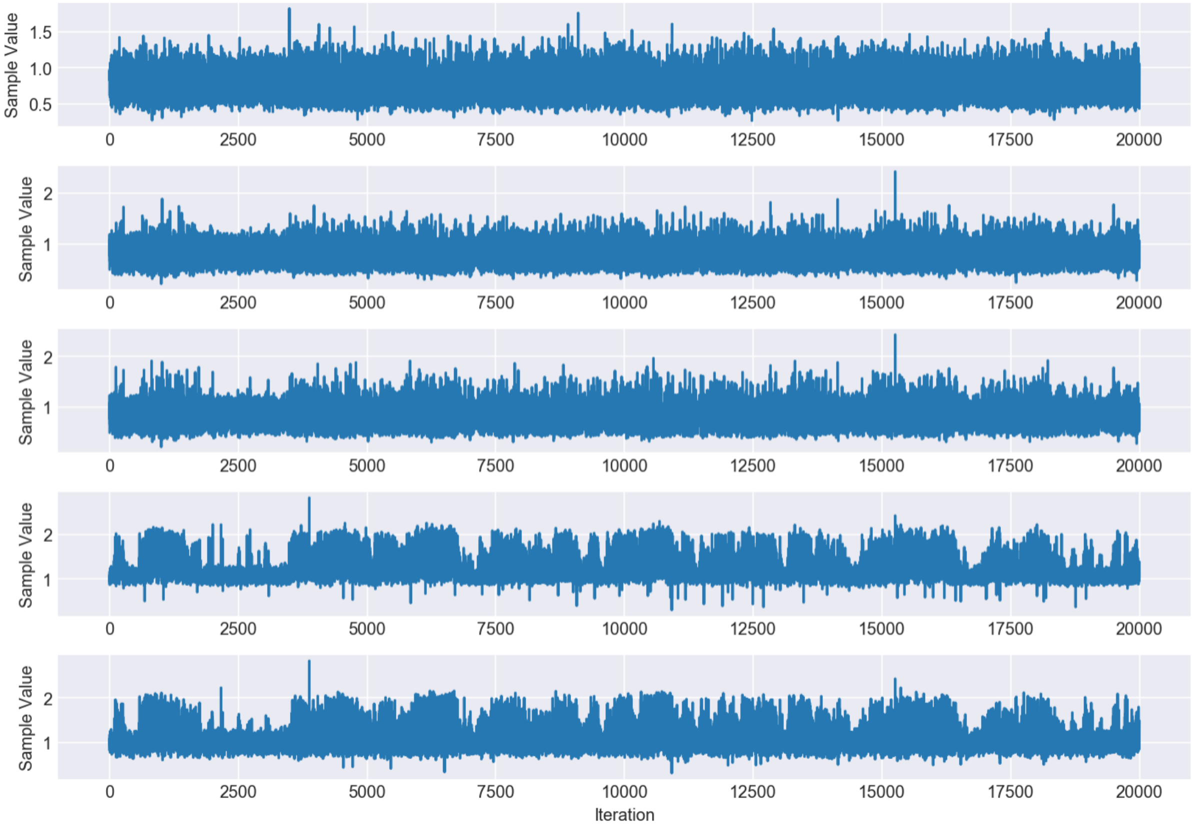

Figure 7: Traceplot for selected pixels within the image after $20\,000$ iterations with a burn-in of 4000 iterations. Pixels $3, 33, 333, 1222$ and 1777 were randomly selected to showcase the convergence of the sampled $x(u_j, v_j)$ values and are displayed in that order from top to bottom. Note that pixels 1222 and 1777 were originally "black" in the true image, where as pixels $3, 33$ and 333 were orginally "white".