Indian Journal of Economics and Development, Vol 5 (9), September 2017 ISSN (online): 2320-9836
ISSN (Print): 2320-9828

Boltzmann and Non-Boltzmann sampling for image processing

T. Pramananda Perumal*, K.R. Srivaishnavi*, D.L. Asha Rani*, K.P.N. Murthy**

**Presidency College, Kamarajar Salai, Triplicane, Chennai-600 005, Tamil Nadu, India*
***Chennai Mathematical Institute (CMI), Siruseri, Kelambakkam, Chennai-603 103, Tamil Nadu, India*
pramanandaperumal@yahoo.com, srivaishnavisuresh07@gmail.com, aancy35@gmail.com, k.p.n.murthy@gmail.com

Abstract

Objectives: We present two algorithms for image processing; the first is based on Boltzmann sampling and the second on entropic sampling.
Methods: These algorithms come within the Bayesian framework which has three components: 1. Likelihood: a conditional density - the probability of a noisy image given a clean image, 2. A Prior and, 3. A Posterior: a conditional density - the probability of a clean image given a noisy image. The Likelihood provides a model for the degradation process; the Prior models what we consider as a clean image; it also provides a means of incorporating whatever data we have of the image; the Posterior combines the Prior and Likelihood and provides an estimate of the clean counterpart of the given noisy image. The algorithm sets a competition between: 1. The Likelihood that tries to anchor the image to the given noisy image so that the features present can be retained including perhaps the noisy ones and, 2. The Prior which tries to make the image smooth, even at the risk of eliminating some genuine features of the image other than the noise.
Findings: A proper choice of the prior and the likelihood functions would lead to good image processing. We need also good estimators of the clean image.
Application: The choice of estimators is somewhat straight forward for image processing employing Boltzmann algorithm. For non-Boltzmann algorithm we need efficient estimators that make full use of the entropic ensemble generated.
Keywords: Image processing, Prior, Posterior, Boltzmann sampling, Entropic sampling, Bayesian.

1. Introduction

We discuss in this paper two algorithms for image processing: one based on Boltzmann sampling (The application of Boltzmann sampling to image analysis was pioneered by Gemen and Gemen [1] and has since become an active field of research [2-6]) and the other on non-Boltzmann sampling (Non-Boltzmann sampling was pioneered by Torrie and Valleau [7]; their method, called Umbrella sampling, has since undergone a series of metamorphoses. We have multi-canonical Monte Carlo algorithm of Berg and Neuhaus [8], entropic sampling of Lee [9], and the algorithm of Wang and Landau [10]. A preliminary and incomplete work [11] on the application of non-Boltzmann sampling to image restoration, indicated that it has no great advantage over Boltzmann sampling). These algorithms are inspired by some recent and not-so-recent developments in Monte Carlo simulation of macroscopic systems. We are presently testing these algorithms on a few benchmark problems employing Monte Carlo simulation. The results shall be presented in a future communication. In this paper, we confine our attention to describing these algorithms and presenting some details on how to implement them.

We begin with a mathematical description of an image, which forms the contents of section (2). This is followed by a brief description of the three basic ingredients of the Bayesian methodology for image processing: the Likelihood distribution (Likelihood function models the process of degradation of a clean image $\hat{\boldsymbol{\theta}}$ to a noisy image X; it is denoted by L(X| $\hat{\boldsymbol{\theta}}$); it is a conditional probability density function: the probability of X given $\hat{\boldsymbol{\theta}}$) [3], in section (3), the Prior (The prior is a probability density function; the prior models what we expect or what we know of the clean counterpart of the given noisy image [4]) in section (4) and the Posterior (Posterior is proportional to the product of the Likelihood and the prior as prescribed by Bayes' theorem; it is a conditional density function P (Θ|X)) in section (5).

We then present the canonical partition function (In the context of statistical mechanics of image processing, the posterior is called the canonical partition function) [5], probability distributions, marginal probability distributions, and threshold posterior mean in section (6). The Metropolis algorithm that generates a reversible Markov chain of images which converges to the desired ensemble is described in section (7). Boltzmann sampling employed to generate a Markov chain which converges to a canonical ensemble is described in section (8). Non Boltzmann sampling that generates micro canonical entropy in the first part of the algorithm and an entropic ensemble in the second part, is described in section (9). We also describe how to calculate the un-weighting factor to estimate micro canonical ensemble averages and re-weighting factor to calculate canonical ensemble averages. In addition we show how to estimate free energy as a function of energy for various values of β. From the free energy profiles we estimate optimal values of the two parameters that lead to minimum free energy. We contend that these optimal parameters would ensure efficient image processing. We conclude the paper with a few remarks in section (10).

2. Mathematical representation of an image

Consider an image plane discredited into tiny squares called pixels. Let *S* be a set of pixels on the digital image plane. For convenience of notation we identify the spatial location of a pixel in the image plane by a single index *i*. Also we identify a pixel by its index *i*. An image can be obtained by painting each pixel belonging to *S* with one of the *Q* gray levels (we do not consider in this paper colour images; however the algorithms described here can be extended to process a colour image [7]); the gray levels are labeled by integers: {0, 1, · · · , *Q* − 1}. Label 0 stands for black and *Q* − 1 for white. If there are *N* pixels in *S*, then in principle we can paint a total of Q^N images. Let Ω denote the image space - a set of all possible images in the image plane.

Mathematically an image Θ ∈ Ω is represented by a collection of integers. Each integer represents the gray level of the corresponding pixel in the digital image plane. The gray level in pixel-*i* is denoted by θ_i. Thus,

$$\Theta = \{\vartheta_i : i \in S\}$$

Let

$$X = \{x_i : i \in S\}$$

Be the given noisy image (Noise enters a digital image by several ways. It can enter while the image is being acquired, due to faulty apparatus or an inexperienced photographer; it can enter while storage due to ageing; it can enter during transmission through a noisy channel; etc[8]). The basic problem of image processing is simple and can be stated as follows: Given a noisy image *X* ∈ Ω how do we obtain its clean counterpart $\hat{\Theta} \in \Omega$?

3. Likelyhood *L*(*X*/Θ)

To answer the question raised at the end of the last section, we first inquire how in the first place the clean image $\hat{\Theta} \in \Omega$ degraded to its noisy counterpart *X*∈ Ω? We do not want to get into the details of how to model the degradation process, except to say that it can possibly be mathematically encoded in an appropriate conditional density called the likelihood function and denoted by the symbol *L*(X|Θ). Note we never know $\hat{\Theta}$, the clean counter part of X. Hence we take Θ as a possible candidate of $\hat{\Theta}$; we ask: if we are given Θ, what is the probability of *X* ? Such a conditional probability density that constitutes the likelihood function is

$$L\,(X|\Theta) \propto \exp\,[-\beta(1-\mu)\,D(\Theta, X)] \qquad (1)$$

In the above $0 < \beta < \infty$ and $0 < \mu < 1$, are constant parameters to be optimized for best image processing.. We shall talk of these parameters later whence their physical meaning would become clear. *D* (Θ, *X*) is referred to as distance function. The degradation of the image Θ to the given noisy image *X* is modeled in the distance function. For example Poisson degradation leads to Kullback-Leibler distance [12], given by

ISSN (online): 2320-9836
ISSN (Print): 2320-9828

$$D(\Theta, X) = (1/N) \sum_i (\vartheta_i - x_i) \ln(\vartheta_i / x_i) \qquad (2)$$

Another useful function that quantifies the separation between two images is called the Hamming [13] distance defined as

$$D(\Theta, X) = (1/N) \sum_i I(\vartheta_i \neq x_i) \qquad (3)$$

In the above, the indicator function is defined as:

$$I(\eta) = \begin{cases} 1 & \text{if the statement } \eta \text{ is true} \\ 0 & \text{if the statement } \eta \text{ is false} \end{cases} \qquad (4)$$

We see that D is just the fractional number of pixels in Θ having gray levels different from those in the corresponding pixels of X. For both the Hamming distance and the Kullback-Leibler distance, it is easily verified that

1. $D(X, Y) \geq 0 \; \forall X, Y \in \Omega$ and equality obtains when $X = Y$ and
2. $D(X, Y)$ is symmetric in its arguments: $D(X,Y) = D(Y, X)$.

In addition the Hamming distance obeys the triangular inequality:

$D(X, Y) + D(Y, Z) \geq D(X, Z) \; \forall X, Y, Z \in \Omega$. The Kullback-Leibler distance, on the other hand, does not obey the triangular inequality. Thus the likelihood which is the probability of X given Θ - tells you how Θ degrades to X due to noise. Bayesian methodology helps us construct the reverse - the probability of Θ given X. To this end the likelihood is multiplied by a suitable prior, see below

4. Prior Π (Θ)

It is one thing to ask for the unknown $\hat{\Theta}$, the clean counterpart of the given corrupt image X and quite another thing to spell out unambiguously what in our opinion constitutes a clean image. Our desire or our expectations about a clean image is encoded in the prior distribution, denoted by the symbol $\Pi(\Theta)$. A good choice of the prior distribution is

$$\Pi(\Theta) \propto \exp[-\beta\mu \, \mathcal{E}(\Theta)] \qquad (5)$$

In the above, $\mathcal{E}(\Theta)$ is called the smoothness function. It measures how smooth an image is. The smoothness function is defined as,

$$\mathcal{E}(\Theta) = (1/N) \sum_{(i,j)} I(\theta_i \neq \theta_j) \qquad (6)$$

The symbol (i, j) in the above equation indicates that the two pixels i and j are nearest neighbours (two pixels that share an edge or a vertex or both constitute a nearest neighbour pair [9]). The sum runs over all the distinct pairs of nearest neighbour pixels. Each pixel in the interior of the image plane has eight nearest neighbours. By employing periodic boundary conditions in both x and y directions, we ensure that even a pixel on the edge or corner of the image plane has eight nearest neighbours. Periodic boundary is equivalent to wrapping the image plane smugly on the surface of a torus.

When all the grey levels are the same, $\mathcal{E}(\Theta) = 0$, and we get the smoothest image possible. Inevitably a realistic image contains features and $\mathcal{E}(\Theta) > 0$. The smoothness function is maximum when each pixel has a gray level different from each its eight nearest neighbours.

5. Posterior $P(\Theta|X)$

Once we have the likelihood and the prior, Bayes' theorem tells you that their product is the desired posterior, $P(\Theta|X)$. The posterior is a conditional probability density function. It is the probability of an image $\Theta \in \Omega$ given the image $X \in \Omega$. More importantly, the posterior sets the stage for a competition between the prior which attempts to make the image smoother and smoother and the likelihood density that tries to keep the image pegged to the given noisy image X.

It is intuitively clear that the images which maximize the posterior stand a good chance of being suitable candidates for the clean counter part of the given noisy image. To appreciate this, let us write the posterior distribution explicitly, see below.

$$\begin{aligned} P(\Theta|X) &\propto L(X|\Theta)\,\Pi(\Theta) \\ &\propto \exp[-\beta\, E(\Theta)] \end{aligned} \quad (7)$$

Where,

$$E(\Theta) = \mu\mathcal{E}(\Theta) + (1-\mu)\, D(\Theta, X) \quad (8)$$

In the language of statistical mechanics, E is the energy of the image Θ. β is $1/[k_B T]$ where T is absolute temperature and k_B is the Boltzmann constant (we set k_B to unity [10]) β is thus inverse temperature. A typical image processing algorithm attempts to locate the region of the image space Ω in which the posterior is maximum. The meaning of μ is now clear.

Its value determines the relative importance given to the prior that drives the algorithm to those regions of Ω containing smooth images and to the likelihood density that tries to keep Θ pegged to X, the given noisy image. μ lies in the open interval between 0 and 1. If $\mu = 1/2$, both "smoothening" and "feature-retention" are given equal importance; if $\mu > 1/2$, the prior is given relatively more importance than the likelihood.

The image is likely to become smooth, perhaps even at the risk of losing some of the genuine features. If $\mu < 1/2$, the likelihood is given more importance. As a result the algorithm tends to retain the features, even the noisy ones. Making the right choice of the value of μ is somewhat difficult. A good strategy, at least for the beginners, is to optimize μ and β by trial and error.

6. Canonical partition function and probabilities

The energy of an image is given by

$$E(\Theta) = \mu\, \mathcal{E}(\Theta) + (1-\mu).D(\Theta, X) \quad (9)$$

We consider a canonical ensemble of images at inverse temperature β characterized by Maxwell-Boltzmann distribution (Called the posterior in the Bayesian framework [11]) given by,

$$P(\Theta) = (1/Q)\exp[-\beta\, E(\Theta)] \quad (10)$$

In the above, Q is called the canonical partition function given by the sum of Boltzmann weight associated with each image, see below.

$$Q(\beta) = \sum_{\Theta} \exp[-\beta\, E(\Theta)] \quad (11)$$

Two possible candidates for $\hat{\Theta}$ are the Threshold Posterior Mean (TPM) and Maximum Posterior Marginal (MPM), defined below.

 ISSN (online): 2320-9836
ISSN (Print): 2320-9828

1. Threshold Posterior Mean (TPM)

We first obtain a set of real numbers $\{\overline{\theta_i} : i \in S\}$, where

$$\overline{\theta}_i = \sum_{\Theta} \theta_i(\Theta)\, P(\Theta) \tag{12}$$

For each *i*, let ζ (i) TPM denote the gray level closest to θ_i. The image constructed with gray

Levels given by ζ (i) TPM d is called the Threshold Posterior Mean (TPM) and is denoted by $\Theta_{TPM.}$ Mathematically,

$$\zeta^{(i)}_{TPM} = \arg \min_{\zeta} (\zeta - \overline{\theta}_i)^2$$

$$\text{for } \zeta = 0, 1, \cdots, Q - 1. \tag{13}$$

$$\zeta_{TPM} = \{ \zeta^{(i)}_{TPM} : i \in S \} \tag{14}$$

2. Maximum Posterior Marginal (MPM)

We partition the image space Ω into mutually exclusive and exhaustive subsets as follows. Consider a pixel $i \in S$. Define $\Omega^{(i)}_{\zeta}$ as a subset of images for which the gray level of pixel-*i* is $\zeta \in [0, Q-1]$:

$$\Omega^{(i)}_{\zeta} = \{\Theta \in \Omega \mid \theta_i(\Theta) = \zeta\} \text{ for } \zeta = 0, 1, \cdots, Q - 1 \tag{15}$$

Calculate now *Q* marginal probability density functions, also called marginal posteriors,

$$\Pi_{\zeta}(i) = \sum_{\Theta \in \Omega^{(i)}_{\zeta}} P(\Theta) \quad \text{for } \zeta = 0, 1, \cdots Q - 1 \quad \forall\, \Theta \in \Omega \tag{16}$$

Define

$$\zeta_{MPM} = \arg \max_{\zeta} \Pi_{\zeta}^{(i)} \tag{17}$$

Which stands the ζ that maximizes the marginal density function $\Pi_{\zeta}^{(i)}$. In other words $\zeta^{(i)}_{MPM}$ is the gray level for which the marginal posterior is maximum. Then the Maximum Posterior Marginal (MPM) image is given by,

$$\Theta_{MPM} = \{ \zeta^{(i)}_{MPM} : i \in S \} \tag{18}$$

We can carry out calculation of TPM and MPM on a set of images generated by suitable Markov Chain Monte Carlo (MCMC) methods based on Metropolis algorithm and to this we turn our attention below.

7. Metropolis rejection algorithm

Consider images characterized by probabilities $P(\Theta)$. Our aim is to generate a large ensemble of images, consistent with the given probabilities. To this end, by employing Metropolis rejection algorithm [14], start with an arbitrary initial image Θ_0 and generate a Markov chain:

Let Θ_i be the current image and $p_i = P(\Theta_i)$, its probability; we make a change in the current image and construct a trial image Θ_t. For example, select a pixel randomly from the current image, and change its gray level to a random value between 0 and $Q - 1$; this operation results in a trial image Θ_t. Let $p_t = P(\Theta_t)$. Calculate

$$p = \min(1, p_t / p_i) \tag{19}$$

Generate a random number ξ uniformly and independently distributed between zero and unity. If $\xi \leq p$ accept the trial image and advance the Markov chain to $\Theta_{i+1} = \Theta_t$. If not, reject the trial image and advance the Markov chain to $\Theta_{i+1} = \Theta_i$. Repeat the process on the image Θ_{i+1}; and iterate. Generate a long Markov chain of images.

8. Boltzmann sampling

Boltzmann sampling obtains when $P(\Theta)$ is given by Eq.(10) which describes a canonical ensemble. Notice that Metropolis algorithm requires only the ratio of probabilities. It is precisely this property that is responsible for the great popularity (Metropolis is considered as one of the great algorithms of the twentieth century [12]) of the Metropolis algorithm.

We carry out Monte Carlo simulation and collect a large sample of images from the asymptotic part of the Markov Chain. We carry out averaging over Monte Carlo ensemble to estimate $\hat{\Theta}$.

We need to choose appropriate values of $\beta \in (0,\infty)$ and $\mu \in (0, 1)$ for processing the given image *X.* Obviously these parameters shall depend on the quantum of noise present in *X* and the relative importance we want to give to the competing criteria of making the given image smooth and retaining its features. A good numerical strategy is to carry out small sample simulations at a set of points in the two dimensional parameter space β and μ. Examine the processed image and find the values of β and μ which give the best image - an image that is smooth and holds all the "relevant" features of *X*.

9. Non-Boltzmann sampling

Non-Boltzmann sampling proceeds as follows [15]. Let G(E)dE denote the number of images having energy in the interval dE around E. Let Θ be an image and $E_\Theta = E(\Theta)$, its energy.
We define an ensemble characterized by the probabilities

$$P(\Theta) \propto 1/G(E_\Theta)' \tag{20}$$

Defined for all the images. Natural logarithm of G(E) is usually called the micro canonical entropy:

$$S(E) = k_B \ln G(E) \tag{21}$$

Where we set $k_B = 1$. We employ Metropolis rejection technique to generate images based on these probabilities. The probability of acceptance of a trial image is thus given by,

$$p = \min(1, p_t/p_i) = \min(1, G(E_i)/G(E_t)) \tag{22}$$

See section (7). Note if the trial image belongs to a low entropy region, it gets accepted with unit probability; if not, its acceptance probability is less than unity. The ensemble generated by the algorithm will have equal number of images in equal regions of energy. In other words, the histogram of energy of the images of the ensemble shall be flat. But a crucial point remains: we do not know the function G(E), as yet.

Wang and Landau [10] proposed to estimate G (E) in an initial learning run. We define a function g(E) and set it to unity for all E. We also define a histogram of energy H (E) and set it to zero for all E. Start with an initial image Θ_0. Let $E_0 = E(\Theta_0)$ is its energy. Update $g(E_0)$ to $\alpha \times g(E_0)$ where α is called the Wang-Landau factor, set to $\alpha_0 = e$, in the first iteration. Also update $H(E_0)$ to $H(E_0) + 1$. Construct a Markov chain of images as per Metropolis rejection technique taking the probabilities proportional to $1/g(E(\Theta))$. Every time you advance the Markov chain, update g and H. Set $\alpha = \alpha_1 = \sqrt{\alpha_0}$, reset $H(E) = 0 \ \forall \ E$ and proceed with the second iteration of the learning run. The value of α tends to unity upon further iterations. After some twenty five iterations, $\alpha = \alpha_{25} \approx 1 + 3\times10^{-7}$.

The histogram would be flat over the range of energy of interest. Flatter the histogram, closer is g(E) to G(E). We take g (E) at the end of last iteration of the learning run that leads to a reasonably flat histogram, as an estimate of G (E). We can define suitable criteria for flatness of the histogram.

We first attach a statistical weight of unity to each image of the entropic ensemble. Then we divide the statistical weight by

$$P(\Theta) = 1 / g(E_{\Theta})' \qquad (23)$$

This is called un-weighting. Upon un-weighting, the ensemble of weighted images for a given energy becomes micro canonical at that energy. In other words, weighted averaging over the images (belonging to a given energy) of the entropic ensemble is equivalent to averaging over a micro canonical ensemble for that energy. We further re-weight to a canonical ensemble at the desired temperature. To re-weight, we multiply the statistical further by $\exp(-\beta E_{\Theta})$. Thus for every image Θ belonging to the entropic ensemble, we have a weight factor given by

$$W(\Theta) = g(E_{\Theta}) \exp(-\beta E_{\Theta}) \qquad (24)$$

These weight factors are used in all averaging operations carried out for estimating $\hat{\boldsymbol{\theta}}$. An advantage of non-Boltzmann sampling is that we can estimate a phenomenological free energy as a function of energy for various values β and μ from the entropic ensemble, see below.

$$F(\beta, E) = -(1/\beta) \sum_{\Theta} g(E'(\Theta)) \exp[-\beta E'(\Theta)]\, I(E'(\Theta) = E) \qquad (25)$$

Where the sum is taken over the images in the entropic ensemble. From the free energy profiles we can estimate $\hat{\boldsymbol{\theta}}$. To this end we need to devise a suitable estimator.

10. Epilogue

We have outlined in this paper two algorithms for image processing: the first based on Boltzmann sampling and the second on non-Boltzmann sampling. We have not yet tested the algorithms on benchmark problems. This work is in progress and preliminary results, not presented here, look encouraging. Non-Boltzmann Monte Carlo algorithms are presently being employed widely in statistical mechanics and we believe these will also prove useful in image processing. We need efficient estimators of clean image to exploit the full potential of an algorithm. For Boltzmann sampling the threshold posterior mean (TPM) and the maximum marginal posterior (MPM) have been found quite effective. For non-Boltzmann sampling we need to test the effectiveness of these estimators in utilizing the rich information contained in the entropic ensemble.

Perhaps we need to innovate and devise new estimators to ensure maximum use of the entropic ensembles. We believe it is worthwhile investigating these and related algorithms and we contend something useful would emerge that would help us process an image better. With this optimistic note, we close this paper.

11. References

1. S. Gemen, D. Gemen. Stochastic relaxation, Gibbs distributions, and Bayesian restoration. *IEEE Transactions of Pattern Analysis and Machine Intelligence*. 1984; 6, 721-741.
2. J.E. Besag. On the statistical analysis of dirty pictures (with discussions). *Journal of the Royal Statistical Society Series B*. 1986; 48(3), 259.
3. E. Tanaka. Statistical mechanical approach to image processing. *Journal of Physics A: Mathematical and General*. 2002; 35(37), 1-81.
4. J.M. Pryce, A.D. Bruce. Statistical mechanics of image restoration. *Journal of Physics A: Mathematical and General*. 1995; 28, 1-511.
5. H. Nishimori, K.M.Y. Wong. Statistical mechanics of image restoration and error correcting codes. *Physical Review*. 1999; 60(1), 132-44.
6. K.P.N. Murthy, M. Janani, B. ShenbagaPriya. Bayesian restoration of digital images employing Markov chain Monte Carlo - a review. ArXiv. 2005.

ISSN (online): 2320-9836
ISSN (Print): 2320-9828

7. G.M. Torrie, J.P. Valleau. Nonphysical sampling distributions in Monte Carlo free-energy estimation - umbrella sampling. *Journal of Computational Physics*. 1977; 23(2), 187-199.
8. B.A. Berg, T. Neuhaus, Multicanonical ensemble: A new approach to simulate first-order phase transitions. *Physical Review Letters*. 1992; 68(9).
9. J. Lee. New Monte Carlo algorithm: entropic sampling. *Physical Review Letters*. Erratum. 1993; 71.
10. F. Wang, D.P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*. 2001; 86.
11. K.V. Ramesh. Boltzmann and non-Boltzmann sampling for image processing, Thesis, M Tech (Computational Techniques), University of Hyderabad. 2009.
12. S. Kullback, R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*. 1951; 22(1), 79-86.
13. R.W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*. 1950; 29(2), 147-160.
14. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machine. *Journal of Chemical Physics*. 1953; 21, 1-7.
15. K.P.N. Murthy. Non-Boltzmann ensembles and Monte Carlo simulation. *Journal of Physics: Conference Series*. 2016; 759.

The Publication fee is defrayed by Indian Society for Education and Environment (www.iseeadyar.org)
***Cite this article as*:**
T. Pramananda Perumal, K.R. Srivaishnavi, D.L. Asha Rani, K.P.N. Murthy. Boltzmann and Non-Boltzmann sampling for image processing. *Indian Journal of Economics and Development.* Vol 5 (9), September 2017.