Inference

James O'Reilly[1], Adam Pluck[2] and Jake Witter[3]

[1] *35055*, [2] *34013*, [3] *35445*

In machine learning, it is often the case that the evidence is computationally or analytically intractable. In these cases, different methods can be used to sample from or approximate our posterior. This report outlines and compares 3 different approaches, in the context of recovering a noisy image. In this case, the original image is our posterior that we are attempting to sample from.

Throughout the report, we use Figure 1 - a grayscale 128 x 128 pixel image - as the clean image that we are trying to recover.

Figure 1: *Haunter*

Question 1

When implementing ICM, a few approaches present themselves, some of which are crude and some of which are more sophisticated. Originally, the implementation of ICM that we chose was quite crude. When classifying a pixel based on its neighbours, we merely counted the number of black or white pixels in the neighbourhood and assigned the pixel to the majority class. There are a few issues with this approach: we do not have the ability to place a weighting on the neighbouring pixels. Furthermore, this approach does not account for the centre pixel from the previous iteration and so only takes into account the neighbouring pixels and not the pixel value itself.

In light of these flaws, we implemented a more sophisticated classification method, based on the energy function defined in Bishop [1].

$$E(\mathbf{x}, \mathbf{y}) = -\beta \sum_{(i,j)} x_i x_j - \eta \sum_i x_i y_i \qquad (1)$$

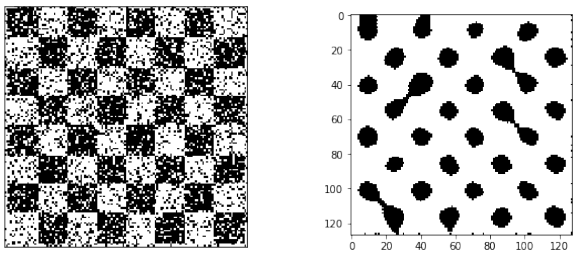

Figure 2: *Results of a poor ICM function on an image of a chessboard after seven iterations. Note how white propagates through the image.*

The β and η parameters allow for a weight to be placed on either the neighbouring pixels or the centre pixel based on the preferences or beliefs about the image. If we had some prior assumptions about the nature of the image, these could be encoded in the ICM function. For example, if we believed the image had mostly horizontal and vertical lines, we would apply more weight to the pixels above and below the current pixel, and weight less heavily in the direction of the diagonals.

Finally, one must decide how to classify a given pixel when entropy is maximum ($H(x) = 1$). If the pixel is classified as white (without loss of generality) then this would cause a white trend to propagate through the chain of dependent probabilities and therefore through the image over time. This behaviour is evident when comparing the images in figure 2. ICM will then terminate once the image is fully white (with the exception of a few edge pixels). Therefore, in the case of max entropy it is best to leave the pixel unchanged from the previous iteration (This is under the assumption that when the image was corrupted, each pixel is flipped with probability < 0.5). This approach ensures that the image terminates in a reasonable state.

Looking at figure 9, it is clear that ICM makes the vast majority of its changes in the first iteration and then quickly reaches a local maximum. In terms of accuracy, ICM fails to maintain the finer details of the image but preserves the general structure rather well while eliminating the salt and pepper noise.

ICM can only optimise the posterior $p(\mathbf{y}|\mathbf{x})$ locally and cannot guarantee that a global maximum will be found. The posterior distribution can be viewed as

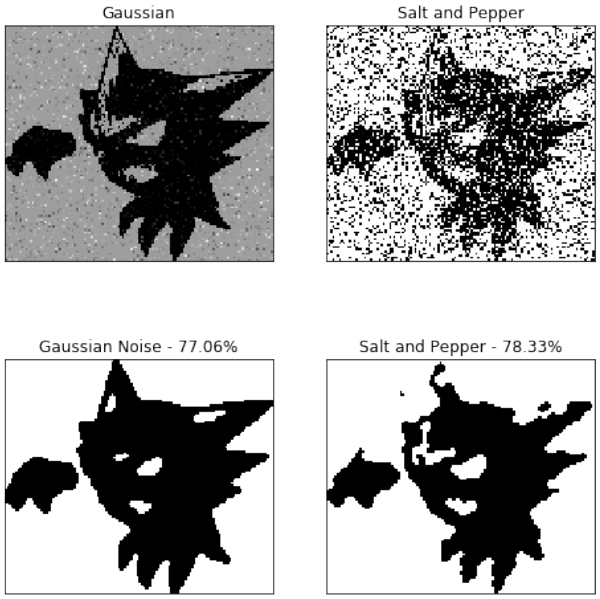

Figure 3: *ICM results for different noises, with percentage match. Prop = 0.2, VarSigma = 0.2*

a manifold. If this manifold has many local maxima, then ICM will be sensitive to initialisation and may perform poorly. An intuitive way to think about this problem is to imagine the movement of a ball on a surface with many sinks, and we want the ball to settle in the deepest sink. Once the ball rolls into one of the sinks (a local minimum) then it cannot leave this local minimum and find the global minimum. The ball will reach different minima dependent on where on the surface it is originally placed. The same problem arises when trying to maximise the posterior in ICM. Random restarts may mitigate the problem and for this reason it may be beneficial to perform ICM by visiting pixels randomly and trying out different initialisation configurations.

Unlike Gibbs sampling, ICM is guaranteed to quickly converge to a local maximum. This is a result of the fact that ICM monotonically increases the estimate of the posterior $p(\mathbf{y}|\mathbf{x})$. Given this fact, and using that the value of this posterior is bounded above by 1, the Monotone Convergence Theorem can be used to prove that ICM will always converge to a local maximum.

Question 2

In an attempt to obtain better results, we implemented the Gibbs Sampling Ising Model. This is an example of a Markov Chain Monte Carlo method. After implementing the algorithm, one must decide how to implement the prior and likelihood term from the equation. The prior belief is simply that the value of a pixel is similar to its neighbours, and the likelihood gives that $\mathbf{y}_i$ and $\mathbf{x}_i$ should be similar.

For the likelihood function, we used Equation 2.

$$p(\mathbf{y}_i|\ \mathbf{x}_i) = \exp(\eta \mathcal{L}_i(\mathbf{x}_i, \mathbf{y}_i)) \tag{2}$$

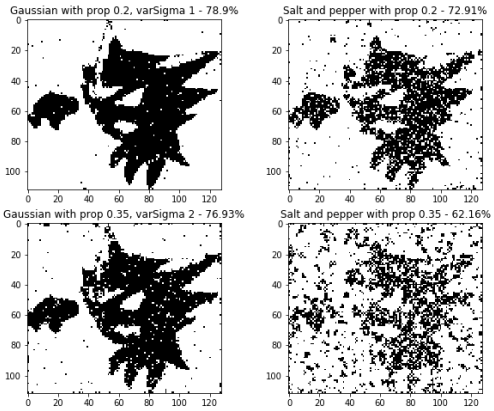

Figure 4: *Gibbs results from noise different levels, with percentage match*

Intuitively, $\mathcal{L}_i$ should be defined so that the function would give a larger value for more similar $\mathbf{x}_i$ and $\mathbf{y}_i$. One such function that satisfies this requirement is

$$\mathcal{L}_i = \frac{1}{|\mathbf{x}_i - \mathbf{y}_i|} \tag{3}$$

As we cannot divide by zero, $\mathbf{x}_i$ and $\mathbf{y}_i$ should never be equal. To avoid this, we use values of 1.1. and -0.1 for $\mathbf{x}_i$ in place of 1 and 0.

The prior used is defined in equation 4.

$$p(x_i = 1, \mathbf{x}_{\mathcal{N}i}) = \exp\left(\beta \sum_{j \in \mathcal{N}_i} 1 * \mathbf{x}_j\right) \tag{4}$$

Figure 4 shows the performance on the images with Gaussian and salt and pepper noise, with different noise levels. It performs much better with Gaussian noise than with salt and pepper noise.

One noteworthy point is that we use 'percentage match to the clean image' as our measure, in an attempt to evaluate the effectiveness of the denoising. However, this doesn't actually seem like a great metric. For example, an all white image would give (in our case) a percentage match of upwards of 50% - we would much rather be evaluating our results with some kind of measure of 'how much a person thinks it look like the original'. Quantifying this is clearly somewhere between very hard and impossible, so percentage match is what we have gone with. For example, in Figure 4, the bottom right image is bordering on unrecognisable but only scores 7% less than the top right, which does look considerably better.

Question 3

Two different methods were implemented to change the order in which pixels are visited. The first approach is to shuffle the order in which each pixel is visited for

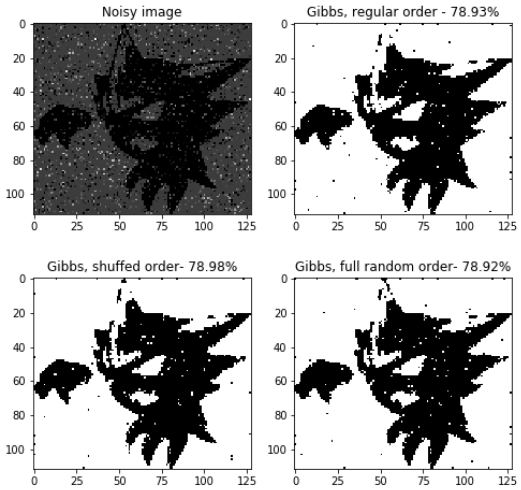

Figure 5: *Results from running Gibbs using different pixel ordering*

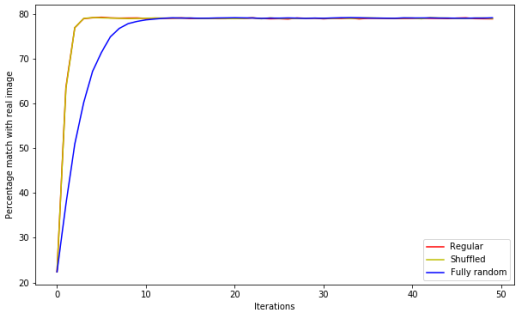

Figure 6: *Graph showing percentage match of different orders for Gibbs applied to our image with Gaussian noise, against iterations*

each iteration. With this approach each pixel is still visited once per iteration. This approach has a negligible effect on the performance of the Gibbs sampler, as can be seen in Figure 6, but in theory removes any systematic bias that results from visiting pixels in a fixed systematic order each time.

The second approach involves fully randomising which pixels are visited. It is therefore extremely unlikely that each pixel is visited each iteration. This does take longer to converge than our other methods, but ultimately converges to the same result, shown in Figure 6.

Figure 5 shows the results of these approaches with their percentage match to the original image. After 50 iterations they give a similar result, and the difference in their matches is effectively negligible and a result of the randomness in the process.

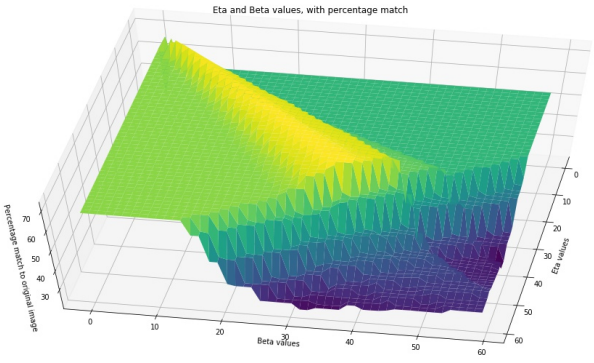

Figure 7: *The effect of varying η and β on our percentage match, rotated to best show*

Question 4

Each approach tends to the same stable point, which in this case, is after approximately 10 iterations. Running past this point yields little or no improvement. The number of iterations to reach this stable point seems to depend on the choice of hyper-parameters - η and β, which also affect the maximum percentage match reached. This problem is then reduced to optimising the hyper-parameters - something that practically is hard, and 'optimised' is dependent upon the bad percentage match metric.

As we have access to the clean image, it is possible to vary the hyper-parameters and see the effect they have on the results, as in Figure 7. The ratio between η and β - that is, the ratio of the weights assigned to the likelihood and the prior - seems like the more important factor, not their exact values. Figure 7 shows how varying these effects Gibbs' accuracy for a set number of iterations and order. However, different images will be 'optimally' denoised by different choices of η and β, thus optimising these parameters with respect to one image is not a good general approach. If this were being implemented in the real world, these hyper-parameters could be learnt from a training set.

Question 5

There are two main types of KL Divergence [4]. First we consider reverse KL Divergence as this is what is use in the model, defined in Equation 5.

$$KL(q(\mathbf{x})||p(\mathbf{x})) = \int q(\mathbf{x}) \log \left(\frac{q(\mathbf{x})}{p(\mathbf{x})} \right) d\mathbf{x} \qquad (5)$$

This shows that the divergence between $q(\mathbf{x})$and $p(\mathbf{x})$is weighted by $q(\mathbf{x})$. This gives most divergence at points where we have a probability mass for $q(\mathbf{x})$and small probability mass for $p(\mathbf{x})$. Clearly if $q(\mathbf{x})$has no probability mass, then the term is zero, and having $p(\mathbf{x})=$ $q(\mathbf{x})$also gives zero. Having low $p(\mathbf{x})$probability mass gives a small denominator, therefore this combined with some $q(\mathbf{x})$mass contributes to divergence. The resulting KL divergence is therefore a measure of how well $q(\mathbf{x})$fits $p(\mathbf{x})$where $q(\mathbf{x})$actually has mass, and does not penalise for sections of $p(\mathbf{x})$for which $q(\mathbf{x})$is zero.

Now consider forward KL divergence - in Equation 6.

$$KL(p(\mathbf{x})||q(\mathbf{x})) = \int p(\mathbf{x}) \log \left(\frac{p(\mathbf{x})}{q(\mathbf{x})} \right) d\mathbf{x} \quad (6)$$

Forward KL weights the integral with $p(\mathbf{x})$, rather than reverse KL which uses $q(\mathbf{x})$. This means that divergence increases anywhere $p(\mathbf{x})$has probability mass, and is increased by low $q(\mathbf{x})$as we are dividing by $q(\mathbf{x})$. This gives large divergence if there are sections of the $p(\mathbf{x})$distribution that $q(\mathbf{x})$does not approximate at all, which is - as you'd expect - the opposite of the reverse case. This measure does permit $q(\mathbf{x})$to have extra probability mass far away from $p(\mathbf{x})$and this does not count towards divergence, which is the reason this is not useful in our case.

Question 6

By using a mean-field approximation, it is assumed that the approximate distribution over each latent variable is independent. This means that each latent variable is entirely parameterised by an independent parameter μ_i. In the case of image inference, one simply needs to find the appropriate values of μ to find a suitable approximating distribution for the pixel's value.

Each of these parameters are calculated using equation 7.

$$\mu_{ij}^{\tau+1} = tanh \left(\sum_{kl \epsilon \mathcal{N}(ij)} \omega_{ij} \mu_{kl}^{\tau} + \frac{1}{2} \left[L_{ij}(1) - L_{ij}(-1) \right] \right) \quad (7)$$

The first term is the sum of the weighted values of the pixels in the neighbourhood. This term is positively large if the neighbourhood is primarily in agreement with the centre pixel, and negatively large if they are primarily in disagreement. This fits with our intuition, as a positive large value evaluated with $tanh$ will give a value of 1, similarly $tanh$ will return -1 when evaluated with a large negative value. When the pixels in the neighbourhood do not belong primarily to one class, then the summation in Equation 7 is much smaller. As a result the parameter μ, when updated, is affected less by the prior.

The second term in Equation 7 is the difference in likelihood of the ijth latent pixel being white or black respectively. This term is large when the observed pixel is white and small when it is black.

In Figure 8, it can be seen that the Variational Bayes method performs exceptionally well for Gaussian noise corruption, but less so for "Salt and Pepper" noise. Experimenting with different values of β and η (in this case weighting the left and right terms respectively) and, one could alter how well Variational Bayes performs with the different noises. One configuration of η and β could perhaps perform well with Gaussian noise and performs poorly with Salt and Pepper noise

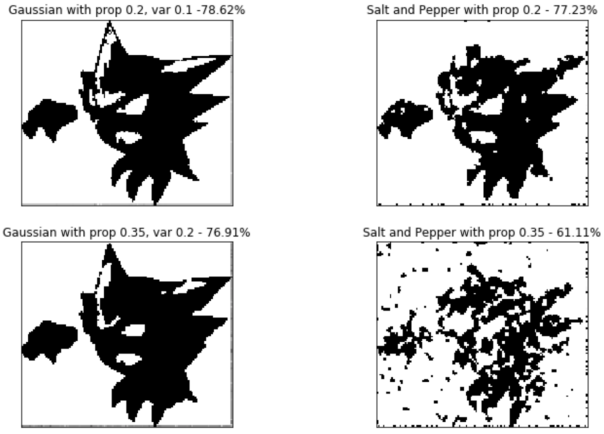

Figure 8: *Variational Bayes by Ising Model*

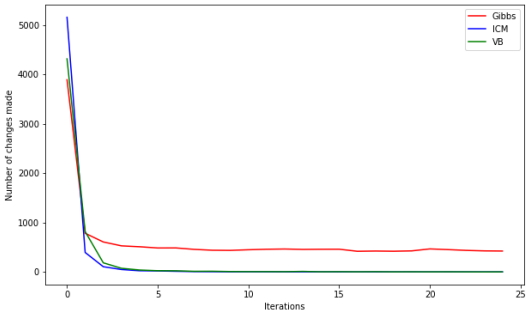

Figure 9: *Number of changes made by each method over iterations*

whilst another configuration could do the complete opposite.

Question 7

So far we have outlined and assessed the performance of the three approaches to image denoising, but only in isolation. Comparing the different approaches allows for assessment of the relative performance of each model, as well as the relative benefits and setbacks. Gibbs sampling is a stochastic approach. Both variational Bayes and ICM are deterministic, however the former is much more sophisticated in its approach. When comparing the different approaches, it is useful to compare under two categories: time taken to converge, and accuracy. The question to ask is whether or not we must sacrifice accuracy in order to achieve efficiency?

Time taken to converge: From figures 9 and 10, we see both ICM and variational Bayes converge faster than Gibbs. It was proved earlier using the monotone convergence theorem that ICM is guaranteed to converge to a local maximum. Convergence of Gibbs is also guaranteed, as shown in this paper [5]. However, Gibbs took much longer to converge than both ICM and variational Bayes.

Accuracy: The accuracy of each approach was mea-

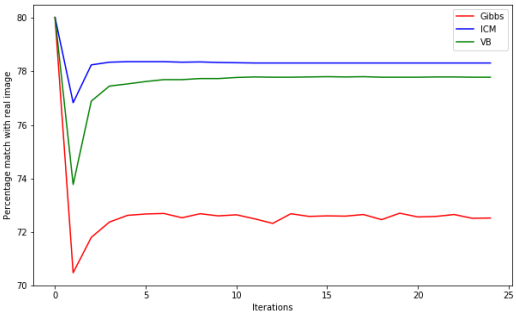

Figure 10: *Percentage match for each method over iterations*

sured by percentage of pixels that matched the original image upon termination. From figure 10 we see that ICM and Variational Bayes out-perform Gibbs when the image is corrupted with salt and pepper noise. However, this graph is based on a partly-flawed metric and does not reflect how good the image may look to a human. To fully understand why Gibbs behaves the way it does, it is useful to understand how it explores the posterior manifold mentioned earlier. Gibbs samples from the posterior and, unlike variational inference, Gibbs is theoretically capable of exploring the entire support of the manifold. The support of a manifold is just those elements which are not mapped to zero (areas that have a non-zero probability mass in our case). Gibbs should then, theoretically, be able to reach a global optimum. However, it would take a long time to do so as Gibbs is computationally expensive. Any modest restriction on the time for which the sampler is running ensures that Gibbs can only explore a localised area of the support and thus cannot find a global optimum.

Variational Bayes, on the other hand, uses approximations of distributions and is thus not capable of finding a global optimum. As a result of this, Gibbs should theoretically out perform variational Bayes if given sufficient run time.

While our data implies that Variational is more accurate than Gibbs over a short time frame, we are skeptical of this result as there is still uncertainty in the literature as to which method is more accurate. The following is an excerpt from a paper by Blei and Kucukelbir and McAuliffe: "Variational Inference: A Review for Statisticians" [2]:// "The relative accuracy of variational inference and MCMC is still unknown. We do know that variational inference generally underestimates the variance of the posterior density; this is a consequence of its objective function. But, depending on the task at hand, underestimating the variance may be acceptable. Several lines of empirical research have shown that variational inference does not necessarily suffer in accuracy, e.g., in terms of posterior predictive densities (Blei and Jordan, 2006; Braun and McAuliffe, 2010; Kucukelbir et al., 2016); other research focuses on where variational inference falls short, especially around the posterior variance, and tries to more closely match the inferences made by MCMC (Giordano et al., 2015)."

Gibbs Sampling or Variational Bayesian Inference? Taking all of this into account, when should one use variational Bayes and when should one use Gibbs, or any other MCMC method? Gibbs is computationally more intensive but can guarantee asymptotically exact examples. Variational Bayes can only give an approximation, but is in general much faster than Gibbs. As variational Bayes can be reduced to an optimisation problem (minimising the KL-divergence), it can make use of optimisation methods, such as those outlined in this paper: [3], to more efficiently approximate the posterior distribution.

The best of both worlds: Given the apparent trade-off that one must make between efficiency and accuracy, it is natural to ask whether or not it is possible to combine the two approaches for better results. This paper [6] titled "MCMC and Variational Inference: Bridging the Gap" (reference paper), outlines the possible ways in which the two approaches could be synthesised to give "fast posterior approximation through the maximisation of an explicit objective, with the option of trading off additional computation for additional accuracy."

Bibliography

[1] Christopher M. Bishop. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN: 0387310738.

[2] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. “Variational Inference: A Review for Statisticians”. In: *Journal of the American Statistical Association* 112.518 (2017), pp. 859–877. DOI: 10.1080/01621459.2017.1285773. eprint: https://doi.org/10.1080/01621459.2017.1285773. URL: https://doi.org/10.1080/01621459.2017.1285773.

[3] Ralf Herbrich. “Minimising the kullback–leibler divergence”. In: (2005).

[4] Agustinus Kristiadi. *KL Divergence: Forward vs Reverse?* https://wiseodd.github.io/techblog/2016/12/21/forward-reverse-kl. 2016.

[5] Gareth O Roberts and Adrian FM Smith. “Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms”. In: *Stochastic processes and their applications* 49.2 (1994), pp. 207–216.

[6] Tim Salimans, Diederik Kingma, and Max Welling. “Markov chain monte carlo and variational inference: Bridging the gap”. In: *International Conference on Machine Learning*. 2015, pp. 1218–1226.