School of Physics
and Astronomy

The Importance of Importance Sampling

Pedagogical Seminar

Emma Grocutt
22 March 2010

Abstract

Importance sampling is a useful technique for investigating the properties of a distribution while only having samples drawn from a different (proposal) distribution. Such methods are often used to estimate a posterior distribution in the framework of Bayesian statistics and thus have important applications in model selection and parameter estimation in the field of astronomy. In this lecture, I will introduce the the concept of importance sampling by explaining in detail one of the most widely–used sampling techniques: the Metropolis–Hastings Markov Chain Monte Carlo. I will discuss the advantages of the technique over traditional grid-based analyses, and highlight some of the considerations that must go into constructing a Markov Chain Monte Carlo. Finally, I will mention some of the limitations of the algorithm and some alternative importance sampling methods that aim to circumvent some of these problems.

Supervisors: Prof. A Heavens, Dr. C Heymans, Dr. T Kitching

1 Introduction

1.1 What is Importance Sampling?

In statistics, importance sampling is the name for the general technique of determining the properties of a distribution by drawing samples from another distribution. On the surface of it, this seems like a strange thing to do — why would we want to sample from a distribution that is different to the one we are investigating? The answer is that the distribution one draws from should be (over a large enough sample size) representative of the distribution of interest, so we can directly infer the desired properties of the second distribution from the first. There are a multitude of techniques for producing such a representative distribution, but before discussing some of these techniques in detail, we shall look at the reasons importance sampling is used in statistical analyses.

1.2 Why is Importance Sampling Important?

Importance sampling is so widely utilised because in many cases in statistical analysis one does not have a direct way of determining the properties of the distribution of interest. This is often the case in parameter estimation or model selection problems where a Bayesian approach is needed (as is standard practice in the field of cosmology). In order to fully appreciate the need for importance sampling in these cases, we must briefly mention Bayes' theorem. For some data D with parameter(s) θ, Bayes' theorem tells us that [1]

$$\pi(\theta)\mathcal{L}(\theta) = E\mathcal{P}(\theta). \tag{1.1}$$

Here $\pi(\theta) = \mathrm{pr}(\theta)$ is known as the *prior probability*, or prior, representing how we originally distribute the parameters' probability ($\mathrm{pr}(\theta)$ denotes the probability of θ). The prior should not depend on the data set D being investigated. One must first decide on the range of the parameters θ which then defines a 'hypothesis space' over which the prior probabilities must be distributed, with the only constraint that the sum of the probabilities is normalised to unity. The choice of prior in a given analysis is often debated, and is usually influenced by data from earlier experiments or observations. In the absence of any previous data to inform our choice of prior, one often uses a 'flat prior' — to assign an equal probability to every region of parameter space. $\mathcal{L}(\theta) = \mathrm{pr}(D \,|\, \theta)$ is the *likelihood*, or probability of the data D occurring given the parameters θ. Often, we can calculate the data value expected from known values of θ, which allows us to calculate the likelihood of the data D for any given point in parameter space. As we will see, the ability to calculate the likelihood in this way proves extremely useful in importance sampling.

The *evidence* is given by $E = \mathrm{pr}(D) = \int \pi(\theta)\mathcal{L}(\theta)d\theta$. This represents how well the priors managed to predict the data, or the average of the likelihood over the whole parameter space. The evidence is the normalising constant in Bayes' theorem and as a result often does not need to be calculated explicitly. Finally, the *posterior probability*, or posterior, is given by $\mathcal{P}(\theta) = \mathrm{pr}(\theta \,|\, D)$. The posterior represents the inferred distribution of probability amongst the models in our parameter space, and it is this distribution that we are seeking to measure. The posterior tells us which models are favoured over others, or which parameter values are a better fit to the data than others. The focus of importance sampling then is to determine as easily

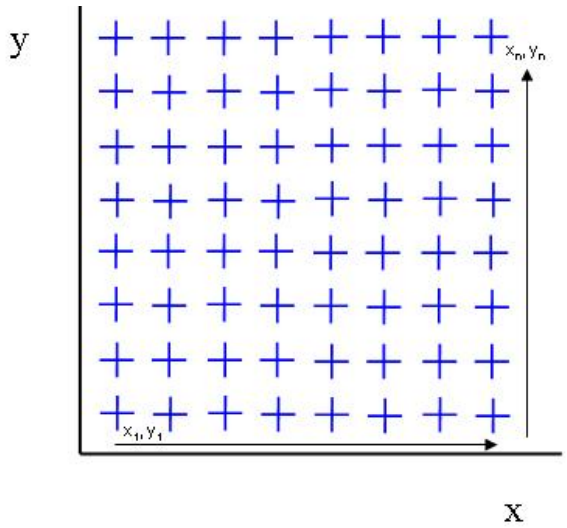

Figure 1.1: Discrete, grid-based sampling of points in parameter space of x and y.

and accurately as possible the properties of the posterior from a representative sample from a second distribution. It is worth noting that with a new data set, the identities of the terms in Bayes' theorem can shift — for example, the posterior derived from one data set can be used as the prior for a new data set.

1.3 Grid-based Analyses

For a given hypothesis or parameter space, the most obvious way to probe the posterior distribution is to use a grid–based approach to calculating the likelihood. Here, one calculates the likelihood of the data D at discrete, evenly–spaced points in the parameter space as shown in Figure 1.1 for the two–dimensional parameter space of x and y. Although we have chosen to vary two parameters in this example for ease of graphical representation, in principle there is no constraint on the number of dimensions N in the analysis. We compare the expected value of D for a given set of model parameters x_1, y_1 with the actual data vector D to find the likelihood at point (x_1, y_1) and repeat for all points through to (x_n, y_n). This will give an idea of the location of the best fit solution and the shape and position of the likelihood contours.

This is a simple way to estimate the posterior, but it has some serious flaws. The main problem with this method is that the number of grid points needed scales exponentially with the number of dimensions (assuming each dimension is the same size) and computation times quickly become prohibitive. A further issue is the nature of gridding itself — how does one decide where to sample the likelihood, and how finely spaced should one's grid points be? We run the risk of missing the 'true' best–fit solution if it lies between grid points.

The gridded likelihood calculation by its nature samples all of the parameter space evenly, meaning much of our computing resources are wasted probing low likelihood regions of the parameter space. Statisticians thus need a tool to help them sample multi–dimensional parameter spaces efficiently while building up a faithful representation of the posterior. Importance sampling provides this tool. In order to understand how importance sampling works, its advantages and its limitations, in the next section we will consider one of the most widely–used examples of the technique: the Metropolis–Hastings Markov Chain Monte Carlo (MCMC).

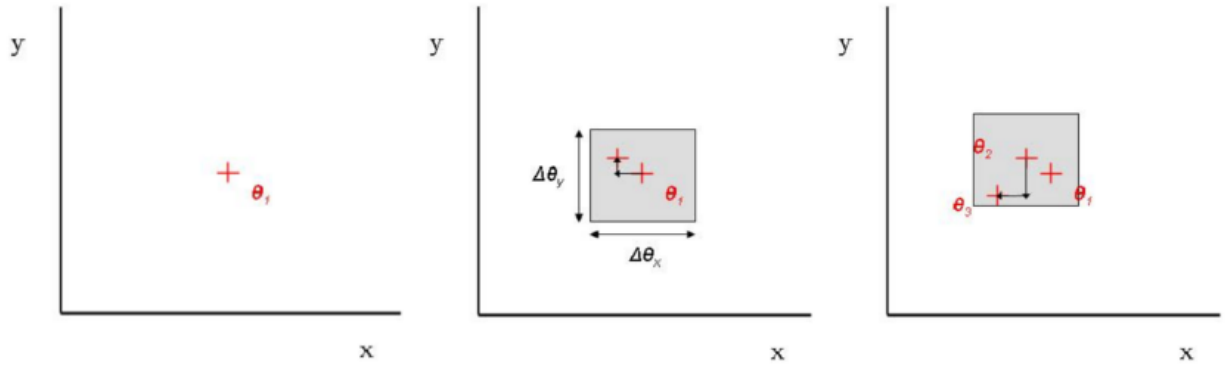

Figure 2.1: *Left*: Starting point θ_1 in x–y parameter space. *Middle*: Stepping randomly from θ_1 to θ_2 within proposal tophat distribution defined by $\Delta\theta_x$ and $\Delta\theta_y$. *Right:* Stepping from θ_2 to θ_3.

2 The Metropolis-Hastings MCMC Algorithm

The Metropolis-Hastings MCMC is a simple yet powerful method for importance sampling. In this section I will go through the construction of the Metropolis-Hastings MCMC algorithm in detail, which I will call MCMC for short. In doing so, I hope to aid in the audience's understanding of importance sampling in general and highlight some of the key issues that need to be considered in its use.

Returning to our x-y parameter space that we considered when discussing gridded likelihood analysis, we define a starting point for our algorithm. For simplicity, we choose a starting point θ_1 randomly anywhere within our x-y parameter space, as shown in the left of Figure 2.1.

We calculate the likelihood of this point in parameter space, then step to a new point, θ_2. This is done by drawing from a proposal distribution around θ_1 and using a random number generator to select a position in the x and y directions for the new point. For simplicity, it is possible to use a box or tophat distribution defined by lengths $\Delta\theta_x$ and $\Delta\theta_y$ as shown in Figure 2.1 (*middle*). The size of the box is decided by the user, and is something that can (and should) be optimised to make the MCMC converge as quickly as possible. We will discuss a technique for finding a good proposal distribution later on. For now, let us choose a box size that is considerably smaller than our parameter space but not prohibitively small.

We then perform the crucial step in the MCMC – we either accept or reject the new point. This is done by calculating the ratio of the likelihoods of θ_1 and θ_2, such that if the new point has a higher likelihood than the old it will be accepted with a probability of 1, otherwise the probability of the point being accepted is equal to the ratio of the likelihoods. Formally this means that

$$P(\theta_i, \theta_{i+1}) = min\left\{1, \frac{\mathcal{L}(\theta_{i+1})q(\theta_{i+1}, \theta_i)}{\mathcal{L}(\theta_i)q(\theta_i, \theta_{i+1})}\right\} \tag{2.1}$$

where $q(\theta_i, \theta_{i+1})$ is the proposal density distribution. For the MCMC, we do not change the

proposal distribution from one iteration to the next so $q(\theta_i, \theta_{i+1}) = q(\theta_{i+1}, \theta_i)$ and therefore

$$P(\theta_i, \theta_{i+1}) = min\left\{1,\ \frac{\mathcal{L}(\theta_{i+1})}{\mathcal{L}(\theta_i)}\right\}. \tag{2.2}$$

It is this selection criterion that gives the Metropolis-Hastings algorithm its name. If θ_2 is rejected, then the chain moves back to θ_1 and selects another point within the proposal distribution at random, calculates the new likelihood and applies the selection criterion again and again until a point θ_2 is accepted. Then the algorithm uses θ_2 as its new starting point and selects another point, θ_3, in the chain from the proposal distribution centred around θ_2, as shown in the right hand side of Figure 2.1. The 'Markov Chain' in MCMC refers to this stepping behaviour; the steps should be discrete, random and the proposal distribution $q(\theta_i)$ a function of current position θ_i only. The ratio of points accepted vs. points tried gives us our acceptance rate. If we are running a chain of n (accepted) points, we want an acceptance rate as high as possible as this will take us to our nth point faster. For example, an acceptance rate of 50% means that we will have to perform $2n$ iterations to achieve a total chain of length n.

The MCMC proceeds in this way until we have built up a collection of points in the parameter space that are sufficient to be used in a likelihood analysis (see Figure 2.2 (*left*)). This is possible because of the elegant fact that the density of points in a given region of parameter space is directly proportional to the likelihood of that region, thanks the the Metropolis-Hastings selection criterion. This ensures that the MCMC is by nature *ergodic*, meaning that any state (point in parameter space) is eventually reachable from any other with a probability of greater than zero. Ergodicity is important because we need to do more than just find the 'best fit' solution — we need to sample the area around a likelihood peak to accurately represent all of the posterior. Thus in principle all of parameter space can be reached and the distribution of our MCMC points should approximate the posterior distribution we are seeking. There are several methods for drawing likelihood contours from the posterior, including using standard χ^2 values to determine contours, however one of the simplest and most reliable ways is to bin the samples on a grid to produce a 2D histogram, and draw contours around the grid points that contain the top (for example) 68% and 95% of points to produce your 68% and 95% contours (see Figure 2.2 *right*). This method works well assuming that the distribution of points faithfully reproduces the posterior distribution. In order for this to be the case, however, there are some additional checks we have to perform.

2.1 Convergence

To ensure that our MCMC chain is both robust and accurate, it must be convergent. This means that the chain has been run long enough to generate a distribution of independent points that closely match the posterior distribution. There is however no single conclusive test that can be run on the results of an MCMC that will tell us if convergence has been reached. One easy test that we can perform is to run multiple chains, each with different random starting points, and compare the results. The variance between the chains should be much smaller than the posterior uncertainty on our measured parameter(s). This idea has been formalised in the form of the *Gelman-Rubin* statistic R =(variance between chains)/(mean variance within the chains). Typically, R should be as close to 1 as possible; preferably $R < 1.03$. MCMC chains will take some time to 'burn-in' (see next section) so the *Gelman-Rubin* test is commonly

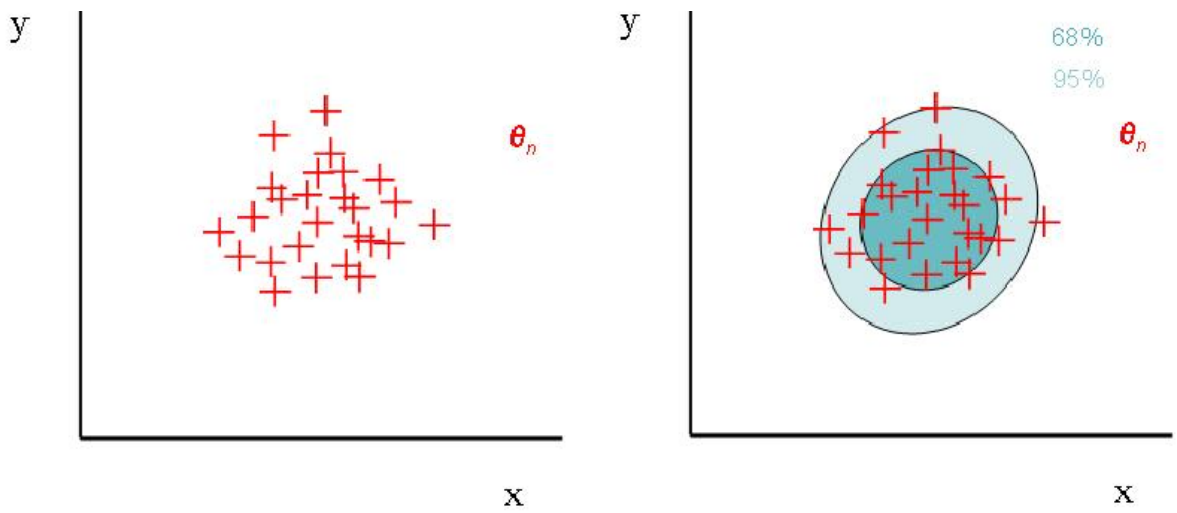

Figure 2.2: *Left*: A chain of points in parameter space from MCMC. *Right*: Sketch of example likelihood contours from MCMC points.

performed on the last half of the points in the chain. Alternatively, we can apply a *Gelman-Rubin* test to a single chain by splitting up the chain and applying the statistic to each part. One can also calculate the correlations between points in the chain as a function of chain length. The distance over which the correlations between parameters drops to 1/e gives a measure of the correlation length, which should be shorter than the chain length if convergence is to be achieved.

The number of chains needed for stability and convergence is not set in stone. It is possible to have several long(ish) chains or one very long chain, as long as either method can pass (multiple) convergence tests. Ideally, we would run both a very long chain and several shorter ones for the same data set and check that both methods give equally reliable results.

2.2 Burn-in

As we have alluded to in the last section, the MCMC chain will suffer from a 'burn-in' at the start of the chain during which the points will not trace the posterior distribution well. This is due to the random nature of the starting point; there is a high chance that the chain will start far from the peak of the likelihood and the first n_b points in the chain will be over–representing a region of lower likelihood. Furthermore, these early points will be correlated (and therefore not independent), as the value of one point will strongly affect the value of the next as they converge towards high likelihood values. For this reason, it is common practice to discard these n_b points from our chain(s) to ensure that they do not bias our likelihood analysis. The question remains as to how large n_b should be made to safely ensure no contamination from burn-in. n_b may be as high as half the entire chain length, but often a much shorter burn-in will suffice. Visual inspection of the data can give us strong clues as to how much burn-in needs to be taken into account. For example, in Figure 2.3 the chain begins to obviously converge on a value of x after about 200 iterations. Finding the optimum burn-in length may thus require some experimentation.

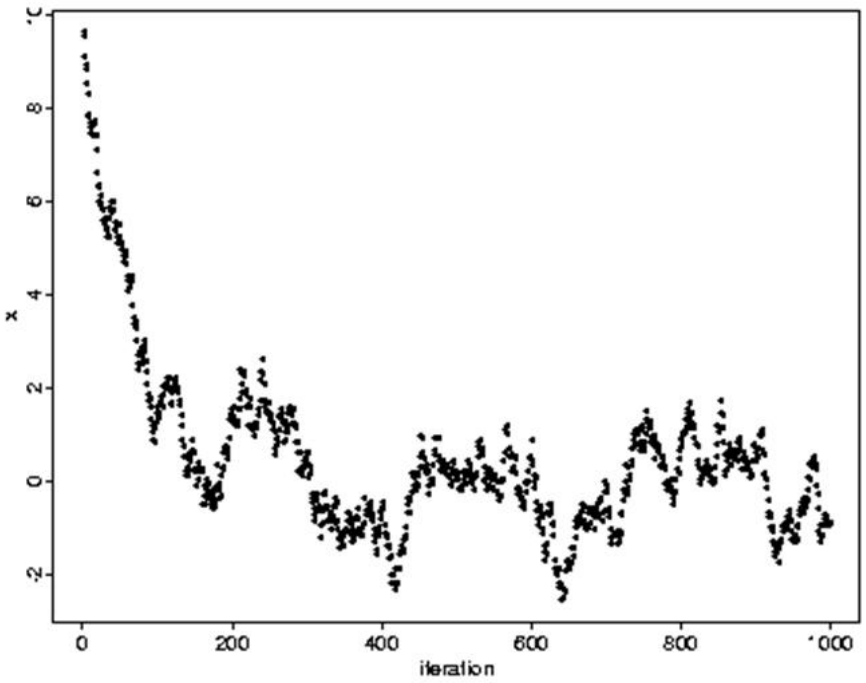

Figure 2.3: Example results from MCMC chain for a parameter x. The burn-in can be clearly seen in the first 200–300 iterations [2].

It is also possible to set up an MCMC to have no burn-in. If we know approximately where the peak of the likelihood surface is (either from knowledge of priors from other data sets or from running a short MCMC 'test chain' to get a feel for the location of the peak first) then we can legitimately select a starting point by centering our proposal distribution on the peak and selecting the first point randomly from this. This potentially makes our chain biased, but the end result (representative sampling of the posterior) is the same [2].

2.3 The Proposal Distribution

In our example above we used a simple box proposal distribution, and intuitively guessed an appropriate size for it from which to draw our points. However, when testing an MCMC one soon discovers that the 'acceptance rate' of the algorithm is strongly dependant on the choice of proposal distribution and hence there is much room for optimisation. We find that the optimal proposal distribution is one which closely matches the posterior distribution. Why should this be so? Consider a situation where we are investigating two parameters that happen to be highly correlated. There may be a strong degeneracy along a certain axis in parameter space, such as that between the cosmological matter density parameter Ω_m and the matter power spectrum σ_8 as shown in Figure 2.4 *left*. Drawing a tophat distribution around a point in the chain will make it hard for the MCMC to step away from this point, as the probability that it will attempt to step to an area of lower likelihood is high. If we are able to select a proposal distribution that closely follows the posterior in both size and shape (in this case, a bivariate Gaussian), this makes it much easier to step to a new point with high likelihood, thus increasing the acceptance rate of the algorithm (see Figure 2.4 *right*).

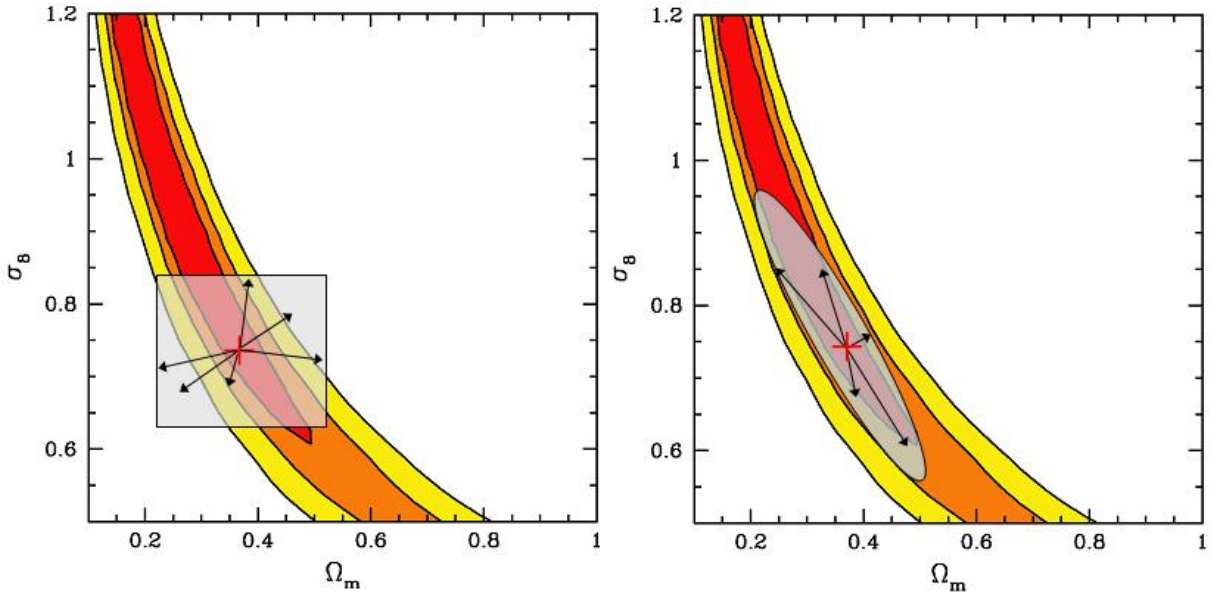

Figure 2.4: *Left*: Tophat proposal distribution gives a poor acceptance rate for the Ω_m–σ_8 degeneracy from weak lensing (Likelihood contours from [4]). *Right*: A bivariate Gaussian distribution results in fewer rejections.

The size of our proposal distribution influences both the acceptance rate and the ease with which a chain converges. A very small proposal distribution will only allow the chain to make tiny steps, which means that although the acceptance rate will be high as nearby points have very similar likelihood values, the chain will take many iterations to explore the parameter space. Thus the chain will take a long time to converge. Conversely, a very large proposal distribution will allow the chain to step easily into different regions of the parameter space, but as soon as the chain finds a point of high likelihood stepping away from it will become very difficult. If $\mathcal{L}(\theta_i)$ is relatively high and θ_{i+1} is located far from θ_i, then $\mathcal{L}(\theta_{i+1})$ will often be much lower than $\mathcal{L}(\theta_i)$ and hence is unlikely to be accepted. Such a chain will therefore also take a long time to run to convergence due to a lowered acceptance ratio. The optimum proposal distribution size is something that must be determined by intuition and experimentation. It is worth noting that since the choice of proposal distribution only influences the acceptance rate, it does not bias the posterior in any way as long as the MCMC is convergent.

If we have good priors on the data and know the expected posterior, it is easy to come up with an adequate proposal distribution. Otherwise, we can run a short chain using a tophat distribution and use the posterior from this to produce our improved proposal distribution for a much longer, convergent run. The proposal distribution itself is often optimised in the form of a (multivariate) Gaussian found from principal component analysis of the posterior[1]. A Gaussian proposal distribution is often a good first order approximation to the posterior, but is not always appropriate and in some cases exploration of the parameter space may still be very slow.

[1]for an explanation of principal component analysis, see for example [3].

2.4 Priors

As we have thus far taken a Bayesian approach to importance sampling, our investigation of the MCMC algorithm would not be complete without a word about priors. For clarification, the priors chosen on x and y in Figures 2.1-2.2 are simply flat and limited by the edges of the parameter space. As stated earlier, this is a common starting point in parameter estimation if no other information is known. The edges of the parameter space may be constrained by limitations in simulations, or bound a region that encompasses all physical solutions.

Often, we may be attempting to constrain our parameters x and y whilst at the same time marginalising over other 'nuisance' parameters. For example we may be interested only in constraining Ω_m and σ_8, but these depend on the values of other cosmological parameters such as the dimensionless Hubble constant h_0. In this case, there are two possible courses of action. One is to let h_0 vary along with Ω_m and σ_8 and then simply ignore the values of h_0 when plotting our likelihood contours - this is marginalisation. The other is to set a prior on the nuisance parameter(s), if possible, from previous data analyses. In this case, we might decide that h_0 has already been well constrained to a value of 0.71 [5] and thus set $h_0 = 0.71$. However, there will most likely be very few samples in our chain for which $h_0 = 0.710000000$ precisely, and it may be better practice to use a narrow prior of $0.70 < h_0 < 0.72$ instead. This is equivalent to assigning a tophat distribution to h_0 over a very small region of parameter space. Adding a little more sophistication to our prior on h_0 might include using the Gaussian error on h_0 and therefore a Gaussian prior for h_0 with a standard deviation of 0.025 [5]. Constraining nuisance parameters using priors will lead to tighter constraints in our likelihood analysis, however we must be confident in our choice of priors to ensure we are not sacrificing accuracy for false precision.

2.5 Caveats

Whilst the number of points needed in grid-based likelihood analyses scales exponentially with dimension number, chain lengths required for convergence in MCMC scale, at best, linearly with dimension number. This is assuming we have an optimised proposal distribution; in reality the scaling will be worse and despite its advantages over a grid-based approach, MCMC can be very slow to converge for high dimensions. In addition, as we have already stated MCMC is not robust to the choice of proposal distribution and this can further slow us down on our quest for convergence.

Another limitation of the MCMC is its difficulty in both probing long tails of a posterior and in dealing with multi-peaked distributions; although the Metropolis-Hastings selection criterion is designed to allow the chain to pass through areas of low likelihood, in practise the MCMC does not sample multiple peaks well. A chain that does not happily sample the full posterior is said to be poorly mixing. One possible solution to a poorly mixing chain is to run multiple chains, each starting in different regions of parameter space in the hope that they will collectively sample all the peaks and regions of the likelihood surface.

Despite its limitations, the Metropolis-Hastings MCMC is a powerful statistical tool if used correctly. A whole host of other methods and algorithms exist, however, in order to overcome some of the limitations of MCMC, and we will briefly mention some of these in the next section.

3 Other Important Importance Methods

3.1 Hamiltonian Monte Carlo

The Hamiltonian Monte Carlo, or Hybrid Monte Carlo (HMC) is a Monte Carlo method that addresses the low efficiency of the Metropolis–Hastings MCMC in high dimensions and its low acceptance rate. In this method, each chain position x_i is randomly assigned a momentum u_i, and we define a potential energy

$$U(\boldsymbol{x}) = -\ln\mathcal{P}(\boldsymbol{x}) \tag{3.1}$$

where $\mathcal{P}(\boldsymbol{x})$ is the posterior or target distribution we are attempting to sample from [6]. As before, we approximate the form of $\mathcal{P}(\boldsymbol{x})$ with our proposal distribution, which may be found in advance from, for example, running a short MCMC chain and performing a principal component analysis on the resulting cloud of points as described in Section 2.3. We can then define the Quantum Mechanical Hamiltonian, $H(\boldsymbol{x}, \boldsymbol{u}) = U(\boldsymbol{x}) + K(\boldsymbol{u})$ where $K(\boldsymbol{u}) = \boldsymbol{u}^T\boldsymbol{u}/2$ is the kinetic energy. This is then used to draw samples from an extended target distribution $\mathcal{P}(\boldsymbol{x}, \boldsymbol{u}) \propto \exp(-H(\boldsymbol{x}, \boldsymbol{u}))$. With our assigned momentum vector, we then follow a trajectory in $(\boldsymbol{x}, \boldsymbol{u})$ phase space, keeping $H(\boldsymbol{x}, \boldsymbol{u})$ constant. The time evolution of the system is governed by the Hamiltonian equations of motion

$$\begin{aligned} \dot{x}_i &= u_i \\ \dot{u}_i &= -\frac{\partial H}{\partial x_i}. \end{aligned} \tag{3.2}$$

In practise the algorithm proceeds by leap–frogging through a series of finite steps in time. One can visualise the likelihood surface of the posterior as a potential well, such that the higher the likelihood value at a given point, the deeper the potential as given by Eqn. (3.1). The time–evolution of the algorithm takes it through a region of constant $H(\boldsymbol{x}, \boldsymbol{u})$ within that surface until a new point in the chain is reached. Then a new, random momentum is assigned and the chain proceeds on another path through phase space to ensure the chain does not get trapped in an ellipse of constant $H(\boldsymbol{x}, \boldsymbol{u})$. In essence, the HMC is an MCMC with a different proposal distribution and a phase space of $2N$ dimensions instead of N due to the presence of the momentum term. To obtain the posterior $\mathcal{P}(\boldsymbol{x})$ after a chain has been run, one simply marginalises over the momentum coordinates in $\mathcal{P}(\boldsymbol{x}, \boldsymbol{u})$ to obtain our desired real–space posterior $\mathcal{P}(\boldsymbol{x})$.

The HMC has the advantage that because the total energy of the system $H(\boldsymbol{x}, \boldsymbol{u})$ is kept almost constant then for two points in the chain, the likelihoods $\mathcal{L}(x_i)$ and $\mathcal{L}(x_{i+1})$ will be almost identical and the acceptance ratio will be close to one. The energy is not perfectly conserved from point to point, however, because of the inexact, numerical nature of the leap–frogging behaviour. Conserving $H(\boldsymbol{x}, \boldsymbol{u})$ also depends on having a full knowledge of the posterior, but of course this is what we are trying to measure with an approximation (the prior distribution). Because the HMC can take relatively large step sizes in parameter space with a high acceptance rate, it can sample the space effectively and without doubling back on itself as the MCMC does due to its random walk nature. Finally, the efficiency of the HMC also scales well with dimension number, so it may be well suited to multi–dimensional analysis.

3.2 Nested Sampling

Nested sampling is a relatively new algorithm [1] and proceeds by generating an array of n points $(\theta_1, \theta_2, ..., \theta_n)$ in the parameter space. The likelihood of each point is calculated, and the point with the lowest likelihood $\mathcal{L}(\theta_i)$ is discarded. Then, a new point θ_i' is generated by taking a random step from one of the other points and is accepted iff $\mathcal{L}(\theta_i')$ is equal to or greater than the likelihood of the discarded point. This ensures that over repeated iterations, the likelihood contours from the n points move progressively inwards towards the peak of the likelihood (hence the name). One calculates the cumulative evidence after each iteration until some stopping criteria have been reached, then the posterior is estimated by weighting each point according to its likelihood width (the width is determined by the distance from a point's nearest neighbour). Nested sampling is a robust technique that copes well with 'difficult posteriors, such as multi–peaked or highly correlated distributions. It also requires less manual tuning than some other importance methods such as the MCMC. Like MCMC, however, it can be slow [7].

3.3 Simulated Annealing

'Simulated Annealing' is so named because of the parallel between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a system such as our parameter space. By analogy, $\mathcal{L} = \exp(-energy)$. For a given energy E and 'temperature', T, a perturbation is added and the change in energy calculated. If the change in energy is negative, the new configuration is accepted; if it is positive, it is accepted with a probability given by the Boltzmann factor $\exp(-dE/T)$. This process is repeated for multiple sampling points, then the temperature is reduced and the process repeated until $T = 0$ is reached and the system has the minimum possible energy (and hence the highest possible likelihood). Simulated Annealing has the advantage that it is good at avoiding become trapped on local minima (likelihood peaks) due to the selection criterion.

4 Conclusion

In this lecture, I have introduced the concept of importance sampling and illustrated its significance in Bayesian model selection and parameter estimation. Importance sampling methods offer significant advantages over grid–based analyses due primarily to their vastly improved computing times. By describing the individual steps that make up one of the simplest and most widely–used importance sampling methods, the Metropolis–Hastings MCMC, I have highlighted the importance of one's choice of priors and proposal distribution. Optimising the proposal distribution will lead to substantial gains in efficiency and computing time, and careful consideration of the priors will ensure more a more accurate posterior approximation. One hopes that the distribution generated from importance sampling will accurately reflect the true posterior, which is why testing for convergence is so important. In addition to the MCMC there are a multitude of other importance sampling methods which include the HMC, nested sampling and simulated annealing, as well as many others which can serve as invaluable tools in parameter estimation.

References

[1] M. Hobson, A. Jaffe, A. Liddle, P. Mukherjee, and D. Parkinson, *Bayesian Methods in Cosmology*. Cambridge University Press, 2010.

[2] C. Geyer, "Burn-in is unneccesary," 1993. `http://www.stat.umn.edu/ charlie/mcmc/burn.html`.

[3] J. Shlens, "A tutorial on principal component analysis," 2009. `http://www.snl.salk.edu/ shlens/pca.pdf`.

[4] H. Hoekstra, Y. Mellier, L. van Waerbeke, E. Semboloni, L. Fu, M. J. Hudson, L. C. Parker, I. Tereno, and K. Benabed, "First Cosmic Shear Results from the Canada-France-Hawaii Telescope Wide Synoptic Legacy Survey," *ApJ*, vol. 647, pp. 116–127, Aug. 2006.

[5] E. Komatsu, K. M. Smith, J. Dunkley, C. L. Bennett, B. Gold, G. Hinshaw, N. Jarosik, D. Larson, M. R. Nolta, L. Page, D. N. Spergel, M. Halpern, R. S. Hill, A. Kogut, M. Limon, S. S. Meyer, N. Odegard, G. S. Tucker, J. L. Weiland, E. Wollack, and E. L. Wright, "Seven-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation," *ArXiv e-prints*, Jan. 2010.

[6] A. Hajian, "Efficient cosmological parameter estimation with Hamiltonian MonteCarlo technique," *Phys. Rev. D*, vol. 75, pp. 083525–+, Apr. 2007.

[7] B. Brewer, "Nested sampling," 2009. `http://www.physics.ucsb.edu/ brewer/nested.pdf`.