Project Report: Statistical Analysis and Monte Carlo Methods

Sourav Sarkar

April 3, 2018

Email: ssarkar1@ualberta.ca
Student ID: 1512784

1 Introduction

We look for the validity of theoretical models by performing experiments. Experimental results give us the information on the validity of theoretical predictions we try to find. In order to perform such validity test, statistical analysis is the essential tool associated with every experiments in physics i.e. from table top experiments to Large Hadron collider (LHC) experiment in Cern. Therefore understanding the framework of statistical analysis and its underlying assumptions are important for every experiments.

However with the advancement of technology experimental apparatus have become more complicated and our size of experimental data also have increased in an enormous scale (in 2015 Cern announced to have stored 75 petabytes of LHC data within last three years [1]). With the complexity of experimental setups, statistical models for analysis also have become more complex over the time. Therefore we need to be careful before applying all the statistical concepts into our data analysis by checking the underlying assumptions for statistical theorems. If we cannot apply any such statistical theorem directly, one powerful computational tool, Monte Carlo method has been developed to perform all the calculations from first principles without taking any assumptions into account. Application of Monte Carlo method is not only bound to statistical analysis. Due to its nature of randomness in the computation, Monte Carlo methods are used in many areas of research in physics, e.g. Monte Carlo method is used in every computational simulation which is now an integrated component of most experiments.

In this report we will briefly summarise the basic statistical tools to analyse the validity of theoretical models from experimental data in section 2 ([2], [3]). In this section, we will also discuss the drawback of analytical statistical analysis and how we can overcome the issue with the help of numerical calculation using Monte Carlo methods. In section 3 and 4, we will then discuss how some important types of Monte Carlo works. And finally in section 5 we will discuss the application of Markov Chain Monte Carlo (MCMC) methods in several areas of physics.

2 p-Value Calculation

In this section we will summarise the components that are needed to perform statistical analysis for any model ([2], [3]). Depending on the analysis, we either accept or rule out the model based on the experimental data we observe. All the discussion for statistical components have been mentioned in the class, so I will briefly summarize the area to set context for our further discussion in the report.

2.1 Summary of Statistical Analysis

Let us assume that we have some experimental observables that can be measured in our experiment, $\vec{x} = (x_1, x_2, \ldots, x_n)$. We construct statistical model based on a probability distribution function (pdf) f for the experimental observables, i.e. all the observables are treated as random variables which follow the probability distribution function. Every statistical model is also associated with some free parameters, say $\vec{\xi} = (\xi_1, \xi_2, \ldots, \xi_k)$. So our probability distribution function is described as function of both the variables, $f = f(\vec{x}, \vec{\xi})$. As we have already mentioned that f is a probability distribution function for the observables, we can impose the following condition for a fixed set of parameters $\vec{\xi}$,

$$\int d^n\vec{x}\ f(\vec{x}, \vec{\xi}) = 1. \tag{1}$$

Our primary objective in the statistical analysis is to understand the parameters based on the measurements in experiment. By understanding these parameters we confirm the validity of theoretical model we follow in our experiment. We perform such analysis by introducing likelihood function which is a function of the parameters we use in our model. So the likelihood function is defined as,

$$L(\vec{\xi}) = f(\vec{x}, \vec{\xi}) \quad (\vec{x}\ \textit{are constant}). \tag{2}$$

Now if we consider all the observables to be independent and follows the above mentioned $f(\vec{x}, \vec{\xi})$, then the joint probability distribution function for the entire data set is given as,

$$f(x_1, x_2, \ldots, x_n; \vec{\xi}) = \prod_{i=1}^{n} f(x_i; \vec{\xi}). \tag{3}$$

For this scenario, the likelihood function becomes,

$$L(\vec{\xi}) = \prod_{i=1}^{n} f(x_i; \vec{\xi}) \tag{4}$$

where we keep all x_is constant and vary the parameters to get the likelihood function values.

Now with the help of likelihood function we can estimate the parameter values in our statistical model by Maximum Likelihood Estimation (MLE) method. In this method we basically look for the maximum value of the likelihood function by

varying the parameter values for a given set of experimental data. The parameter values corresponding to the the maximum likelihood value give us the set of estimated parameters, denoted as $\hat{\xi}$.

The above statistical analysis can be understood with an example. For a particle collider experiment we can measure cross section σ by counting the number of scattered particles that pass certain trigger and filtering criteria. With the understanding of theoretical model with some parameter $\vec{\xi}$, we can write the predicted cross section as $\sigma(\vec{\xi})$. By measuring the cross section in the experiment and using the data for above MLE method, we can estimate the parameters associated with the model.

After the parameter estimation, our next objective is to check the validity of the model we use in analysis. Let us denote the maximum likelihood value i.e. probability distribution function for the estimated parameter as,

$$f^{max}(\vec{x}) = f(\vec{x}, \hat{\xi}). \tag{5}$$

Now we consider a constrained model by fixing some of the parameters of the model we have introduced. For this constrained model, we again estimate the free parameters which is denoted as $\hat{\xi}_c$ and its corresponding pdf is denoted as,

$$f_c^{max}(\vec{x}) = f(\vec{x}, \hat{\xi}_c). \tag{6}$$

As we have defined maximum likelihood values for both the models, we can now define the test statistics t as follows,

$$t(\vec{x}) = -2ln\left(\frac{f_c^{max}(\vec{x})}{f^{max}(\vec{x})}\right). \tag{7}$$

With the test statistics defined, we perform our statistical analysis by setting two hypothesis to compare - null hypothesis and alternative hypothesis. We set our null hypothesis by estimating the parameters from actual experimental data, say $\vec{x}_0$ with MLE method for constrained model. If we denote such estimated parameters as $\hat{\xi}_0$, then our null hypothesis is denoted as, $f_c(\vec{x}, \hat{\xi}_0)$ and the unconstrained model becomes our alternative hypothesis denoted by the pdf $f(\vec{x}, \hat{\xi})$. We then perform the statistical test by calculating statistical significance of the null hypothesis with p-value defined as,

$$p = \int d^n\vec{x}\ f(\vec{x}, \hat{\xi}_0)\ \theta(t(\vec{x}) - t(\vec{x}_0)) \tag{8}$$

From the definition in Eq.8, we see that p-value is basically the area under the pdf for null hypothesis in a certain region set by the θ function. The region is defined by setting the boundary with observed data point and is valid for all the test statistics which is greater than the boundary value. The definition can also be understood with the help of fig. 1.

So p-value defines the probability of obtaining our experimental result under the null hypothesis assumption. The probability assumes that we expect to see our experimental result at the boundary point or even far away from the null hypothesis. Therefore with smaller the p-value our experimental result will have larger

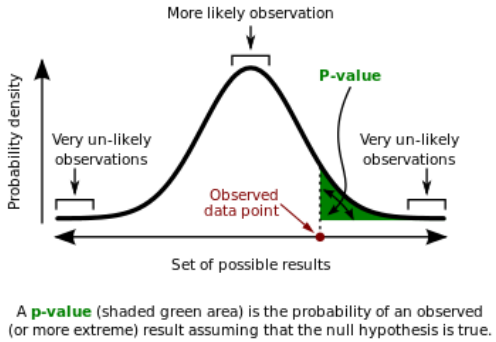

Figure 1: Schematic diagram to show p-value calculation region. The black solid line represents the null hypothesis distribution. Dotted vertical line with a dot in x-axis represents the boundary set by observed data and green shaded region represents the p-value. This figure is taken from [4].

significance. In different field of physics we set different significance level for the acceptance of our experimental result for underlying model. For example, in particle physics the significance level for acceptance is set at 5σ. So we see p-value calculation is an essential tool for any statistical analysis of experimental result.

2.2 p-Value Calculation Methods

In simple cases for large number of data set calculation of p-value is trivial. We achieve this calculation by following Wilk's theorem [5]. Wilk's theorem states that for such large number of data set, test statistics from Eq.7 follows usual χ^2 distribution with degrees of freedom being the difference between the dimensionality of ξ and ξ_0. Using this Wilk's theorem we can write the p-value integral in the following analytical form [6],

$$p = 1 - \Gamma_{\nu/2}(t(\vec{x_0})/2) \tag{9}$$

where Γ is the normalised lower incomplete gamma function and $\nu = dim(\xi) - dim(\xi_0)$.

As an application, for global analysis in particle physics, calculation of p-value is performed using Wilk's theorem. However there are some underlying assumption for using Wilk's theorem, which are the following,

- Models used in the analysis are nested i.e. constrained model we have used in our p-value formulation is a subset of the unconstrained model we started with.
- Maximum likelihood estimation of parameters for experimental observables $\hat{\xi}(\vec{x})$ follows a Gaussian distribution.
- The parameter spaces of $\vec{\xi}$ and $\vec{\xi_c}$ are perfect hyperplanes in their respective dimensions.

So if we use Eq.9 in our analysis, we automatically follow above assumptions for our model. However, in reality this assumptions are not exact but mere approximations for Wilk's theorem. In cases where analysis result is sensitive in terms of significance level and these assumptions are not satisfied exactly, we might calculate p-value incorrectly by using Wilk's theorem which can affect the entire analysis.

One possible solution to this problem is to calculate p-value numerically without assuming Wilk's theorem. So we use the fundamental definition of p-value in Eq.8 and perform numerical integration. We can perform numerical integration deterministically using methods like trapizoidal rule, Simpson's rule etc. However this methods works better only with 1-D integrations. For higher dimensional numerical integration, we mostly use Monte Carlo methods. One popular Monte Carlo method is importance sampling Monte Carlo. Understanding the working principle of such Monte Carlo method is important as we see its role in statistical analysis and in other application. In the next sections we will now concentrate on how these Monte Carlo methods work.

3 Ordinary Monte Carlo Methods

Monte Carlo methods span a broad range of computation algorithms where we perform numerical calculation using randomness of sampling which follows some specific distribution. In this section we will discuss about two popular Monte Carlo methods: importance sampling and rejection sampling Monte Carlo.

3.1 Importance Sampling Monte Carlo Method

Suppose we want to perform an integration $\int f(x)dx$ numerically where the variable x follows specific probability distribution function (pdf) $p(x)$. So considering the variable distribution, the integral becomes,

$$I(f) = \int f(x)\ p(x)\ dx. \tag{10}$$

To perform this integration numerically, we draw samples from the distribution $p(x)$ and then integrate the function $f(x)$ in the following way,

$$\hat{I}_N(f) = \sum_{i=1}^{N} f(x^i) \tag{11}$$

where the variable sample x^i is drawn from the pdf. However in cases where drawing sample from $p(x)$ is difficult, we introduce another distribution called *proposal distribution* and it is denoted as $q(x)$. We choose this proposal distribution such a way that drawing sample from this distribution is relatively easy. Then we rewrite the integral in Eq.10 in following way,

$$I(f) = \int f(x)\ p(x)\ dx = \int f(x)\ \frac{p(x)}{q(x)}\ q(x)\ dx \tag{12}$$

where we denote $\frac{p(x)}{q(x)}$ as $w(x)$ which is called importance weight. Now if we can draw N samples from the probability distribution $q(x)$, then we can write the numerical integration form as[7],

$$\hat{I}_N(f) = \sum_{i=1}^{N} f(x^i)w(x^i) = \sum_{i=1}^{N} f(x^i)\frac{p(x^i)}{q(x^i)}. \tag{13}$$

So even if we follow proposal distribution for the integration variable, we incorporate importance weight for each sample according to our target probability distribution function $p(x)$. Therefore this method is called importance sampling Monte Carlo method.

Even though we are taking weights for each sampling, we are finally drawing sample from a different distribution. So calculating the function f in this approach is not exactly the same as the approach where we draw sample from $p(x)$. Therefore there will always be a finite variance in the numerical integration using importance sampling method. The primary objective of using this method is to find the best possible proposal distribution $q(x)$ to reduce the variance i.e. searching for $q(x)$ which will be as close to $p(x)$ as possible.

3.2 Rejection Sampling Monte Carlo Method

Another popular ordinary Monte Carlo is rejection sampling method. We use this Monte Carlo method where drawing sample is important rather performing numerical integration. Let us assume that we want to sample according to a pdf $p(x)$ which is difficult to draw with direct sampling procedure. Therefore we introduce the proposal distribution $q(x)$ in a similar way where drawing sample is easy. In this method we choose the proposal distribution in a way so that it includes the entire $p(x)$ distribution. If necessary we scale up the proposal distribution by multiplying a constant value $M < \infty$ so that the condition $p(x) \leq Mq(x)$ is satisfied. Then rejection sampling algorithm works in following way [7],

- Draw sample x^i from $q(x)$ and also draw u from uniform distribution $U_{(0,1)}$.
- If $u < \frac{p(x^i)}{Mq(x^i)}$, then accept the sample.
- Otherwise reject the sample.
- Repeat.

To understand how this method works, let us consider fig.2 where the single peak Gaussian-like curve follows $Mq(x)$ distribution and double peak curve follows $p(x)$ distribution. If we perform a specific sampling x^i and u in large number then the points $uMq(x^i)$ will be distributed uniformly along the vertical line shown in the figure. Then for the uniformly distributed points, we accept only those which satisfies the condition $uMq(x^i) < p(x^i)$ and reject other points. This procedure can be compared with flipping a biased coin where the probability of getting 'head' is $\frac{p(x^i)}{Mq(x^i)}$ and probability of getting 'tail' is $\frac{Mq(x^i)-p(x^i)}{Mq(x^i)}$. We then count the samples whenever we get heads by flipping the coin. If we perform this procedure for all the

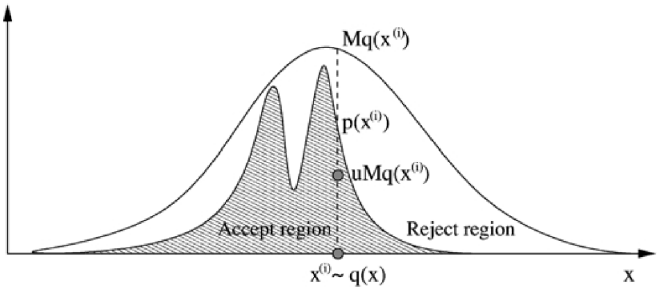

Figure 2: Schematic diagram for rejection sampling method. Single peak distribution represents proposal distribution and double peak distribution represents target distribution. Shaded area is the accepted region in rejection sampling algorithm. This figure is taken from [7].

sample points, then for large number of iterations we eventually end up having the sample distribution similar to the desire pdf $p(x)$.

3.3 Problem with Ordinary Monte Carlo Methods

Although these models work well for simple distributions, there are several drawbacks of rejection sampling for drawing sample. First, it is not always possible to bound the target distribution $p(x)$ with good choice of M as the probability of acceptance goes as $1/M$. If we end up with a large M value in order to bound the target distribution, the acceptance probability becomes too small. Thus for exponentially large sample set, if we have sample with exponentially small probability, we have to wait forever for the sample acceptance. Therefore rejection sampling is inefficient in drawing sample from higher dimension probability distributions. For such scenarios, we move the use of Markov Chain Monte Carlo method which we will discuss in the next section.

4 Markov Chain Monte Carlo (MCMC) Method

Basic objective of Markov chain Monte Carlo is similar to rejection sampling i.e. let us assume we have sample set $X = \{x_0, x_1, \ldots, x_n\}$ and we want to draw sample from this set according to the distribution function $p(x)$. Our goal is to construct an efficient algorithm which outputs sample x^i with probability $p(x^i)$.

4.1 Definition and Properties of MCMC

'Markov chain' part in Markov chain Monte Carlo is essentially a random walk with some constraints and special conditions. We will try to understand this random walk in terms of graph theory. Let us assume we have a graph $G(V, E)$ where V is the set of all the vertices and E is the set of all the edges (lines connecting the vertices) in the graph. We can see one example graph in fig.3. To form Markov chain we implement constrain on the graph that the graph has to strongly connected. A graph is called

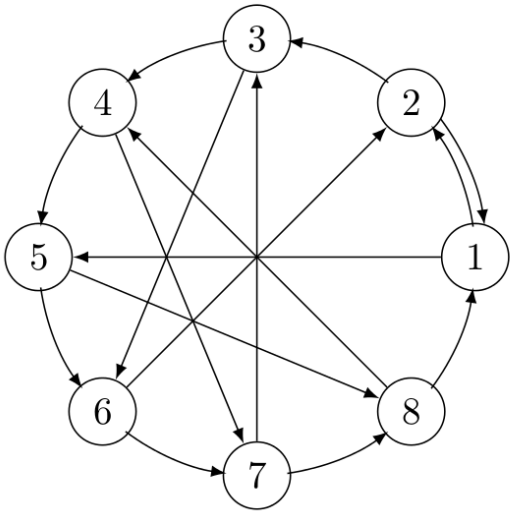

Figure 3: Pictorial representation of a strongly connected graph.

connected when there is a path for every pair of vertices i.e. there will not be any vertices left without any edge. And a graph is called strongly connected when the edges have directions e.g. in fig.3 the edge for going from vertex 1 to vertex 2 is distinguishable from the edge for going from vertex 2 to vertex 1.

After constructing such graph, we assign every edge with some edge probability denoted as $p_{u,v} \in (0, 1)$. This probabilities are also called transition probabilities. These edge probabilities have to satisfy the standard probability normalization condition $\sum_{All\ v} p_{u,v} = 1$ i.e. sum over all outgoing edge probabilities has to be 1 for all vertex u. We notice that the edge probabilities only depend on the last vertex u, not on any other vertex in the graph. This is one important property which builds the chain formation in Markov chain. With this implementation Markov chain is defined as the set $(V, E, p_{e\in E})$ i.e. sets of all the vertices, edges and their respective edge probabilities.

Now let us look at how we can perform random walk in the Markov chain we have defined. We start at some vertex x_0 on the graph. Then we choose one outgoing edge based on all the edge probabilities assigned from vertex x_0. After choosing the edge, we move to the next vertex, say, x_1. Then we repeat the procedure again. This way we walk randomly on the graph according to edge probabilities (transition probabilities) assigned in each step.

With current description of random walk in Markov chain, there is an associated theorem called Stationary Distribution theorem (it is also called fundamental theorem of Markov chain). Stationary distribution theorem states that for a very long random walk in the Markov chain, the probability that we land on some vertex v is independent of where we started. Such probabilities for all the vertices together are called Stationary Distribution of random walk, uniquely defined by Markov chain. Therefore our objective in Markov chain Monte Carlo method is to construct such stationary distribution as our probability distribution function $p(x)$.

4.2 Implementation of MCMC Method

With the description Stationary Distribution theorem, we will now try to construct the stationary distribution of the Markov chain mathematically. Let us consider a matrix A for which we define the matrix element as, $a_{ji} = p_{i,j}$ where $p_{i,j} \in \{p_e\}_{e \in E}$. In cases where there is no edge between any two vertices i, j, we simply put that element to be 0. With this definition of the matrix element, the matrix A has the following form,

$$A = \begin{bmatrix} p_{0,0} & p_{1,0} & p_{2,0} & \cdots & p_{n,0} \\ p_{0,1} & p_{1,1} & p_{2,1} & \cdots & p_{n,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{0,n} & p_{1,n} & p_{2,n} & \cdots & p_{n,n} \end{bmatrix}. \tag{14}$$

If we look carefully, we notice that any ith column of the matrix represents the distribution of all outgoing edge probabilities from vertex i. This helps us to think about the matrix in terms of linear algebra. If we choose any basis vector e_i which has non-zero ith element, the vector looks like,

$$e_i = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1(ith) \\ \vdots \\ 0 \end{bmatrix}. \tag{15}$$

This can be interpreted as the starting of the random walk at vertex i. Now if we apply the matrix A on the basis vector, we get,

$$Ae_i = \begin{bmatrix} p_{0,0} & p_{1,0} & p_{2,0} & \cdots & p_{n,0} \\ p_{0,1} & p_{1,1} & p_{2,1} & \cdots & p_{n,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{0,n} & p_{1,n} & p_{2,n} & \cdots & p_{n,n} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1(ith) \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} p_{i,0} \\ p_{i,1} \\ \vdots \\ p_{i,n} \end{bmatrix}. \tag{16}$$

From Eq.16 we see that any jth element of the final vector gives $p_{i,j}$ which is the probability of of random walk being at vertex j after one step. Through the same procedure if we apply the A matrix n times $A^n e_i = z$, then the jth element of the vector j gives us the probability of being at vertex j after n steps of random walk.

Now if we revisit the stationary distribution theorem mathematically, it is stated as follows:
Let $G(V, E)$ be a strongly connected graph with associated edge probabilities $\{p_{e \in E}\}$ forming a Markov chain. Staring with a probability vector x_0, define $x_{t+1} = Ax_t$ for all $t \geq 1$, and let v_t be the long time average $v_t = \frac{1}{t} \sum_{s=1}^{t} x_s$.Then,
- There is a unique probability vector π such that $A\pi = \pi$
- For all x_0, the limit $\lim_{t \to \infty} v_t = \pi$.

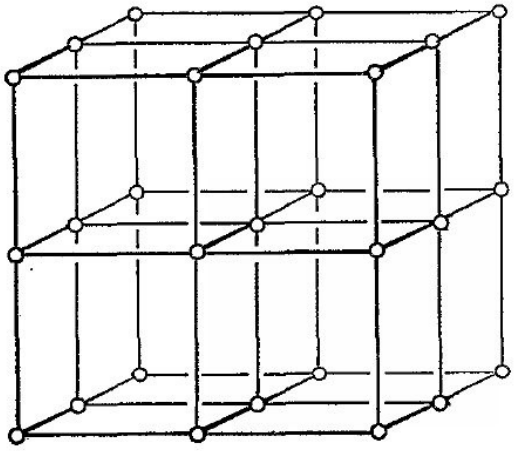

Figure 4: Lattice structure representation of the set $\{0, 1, 2, \ldots, n\}^d$ where $n = 2$ and $d = 3$.

As we have mentioned before, we need to construct Markov Chain such a way that the stationary distribution π vector becomes the desired probability distribution $p(x)$. Therefore if we are able to construct Markov chain, our goal is to find the vector π. It can be done in two ways. We can actually take the average of all the vectors we get in each step of a long random walk (i.e. by calculating the vector v_t as mentioned in the theorem). We can also search for the eigen vector of the matrix A with the help of an eigen solver.

In practical situation we do not use any of the two procedure discussed above because of their inefficiency. So we use an alternative approach to use Markov chain Monte Carlo method. In this procedure, we choose a graph and construct edge probabilities such a way that after long enough random walk we have the stationary distribution as our probability distribution function $p(x)$. Then we draw sample from the set X based on whatever state comes up in the random walk according to stationary distribution. There are several computational algorithm which use this procedure in Markov chain Monte Carlo. The most popular ones are Metropolis-Hastings algorithm [8] and Gibbs sampling algorithm. These are also slice sampling algorithm, reversal jump etc. which performs Markov chain Monte Carlo. In the next section we will discuss how Metropolis-Hastings algorithm works.

4.3 Metropolis-Hastings (MH) Algorithm

We use MCMC method by first constructing a graph G to random walk on. Our popular choice for constructing such graph is to use lattice structure. We construct the lattice structure for a sample set X as $\{x_0, x_1, \ldots, x_n\}^d$ where d is the dimension of the lattice. One such example lattice structure is shown in fig.4 for set $\{x_0, x_1, x_2\}^3$. Here we take one special for of the lattice that all the edges connecting the lattice points are bi-directed with equal probabilities, i.e. probability of going from vertex i to vertex j is same as going from j to i. If we work with this special case of the graph in MH algorithm it is called Metropolis algorithm. Also note that as the bi-directed edges have same probabilities, the A matrix we

have defined before will be symmetric along the diagonal. For the lattice structure described, maximum degrees of freedom for each vertex is $D = 2d$ where d is the dimension of the lattice. Here degrees of freedom denotes the connectivity of each vertex with every other vertices. So clearly, by increasing the dimension d we can increase the vertex connectivity, but it will reduce our computation efficiency. So we have to find one optimized dimension to work with in MCMC.

Recall that our objective is to draw sample with pdf $p(x)$. In Metropolis-Hastings algorithm we again introduce a proposal distribution $q(x|y)$ from which we can draw sample easily.We use this distribution to choose the next vertex for the random walk in Markov chain. For this reason, the proposal distribution is also called jumping probability in MCMC. The proposal distribution $q(x|y)$ to select next vertex x only depends on the vertex y which preserves the chain property of Markov chain discussed before.

With the target pdf $p(x)$ and proposal distribution $q(x|y)$, Metropolis-Hastings algorithm works with the following steps:

- Initialize the random walk by starting at some vertex i_0
- Choose the next sampling point (vertex) from $q(j|i)$. One usual choice of the jumping probability is Gaussian distribution centered at vertex i in each step of the random walk.
- If $p(j)q(i|j) > p(i)q(j|i)$, deterministically go to j: $p_{i,j} = 1$. This implies that if the probability of staying at j is greater than staying at i, random walk will deterministically move to vertex j.
- If $p(j)q(i|j) < p(i)q(j|i)$, go to j with probability $p_{i,j} = \frac{p(j)q(i|j)}{p(i)q(j|i)}$. As $\frac{p(j)q(i|j)}{p(i)q(j|i)} < 1$, for rest of the probability random walk will reject the sample j, i.e.it stay at vertex i instead of moving to j.
- Actual probability of staying at vertex i is $p_{i,i} = 1 - \sum_{(i,j)\in E; j\neq i} p_{i,j}$.
- Repeat the same for next step in the random walk.

In our lattice structure discussion before, as we mentioned the special symmetric case, the jumping probability will be same for both direction, i.e. $q(i|j) = q(j|i)$. With this process randomly walk and draw sample from the distribution.

To check if this algorithm actually produces the stationary distribution as our traget distribution, we can start with the reversibility relation,

$$p(i)q(i|j) = p(j)q(j|i) \tag{17}$$

and fixing i and summing over j, we get,

$$\sum_j p(i)q(i|j) = \sum_j p(j)q(j|i) \tag{18}$$

$$p(i) = \sum_j p(j)q(j|i) \tag{19}$$

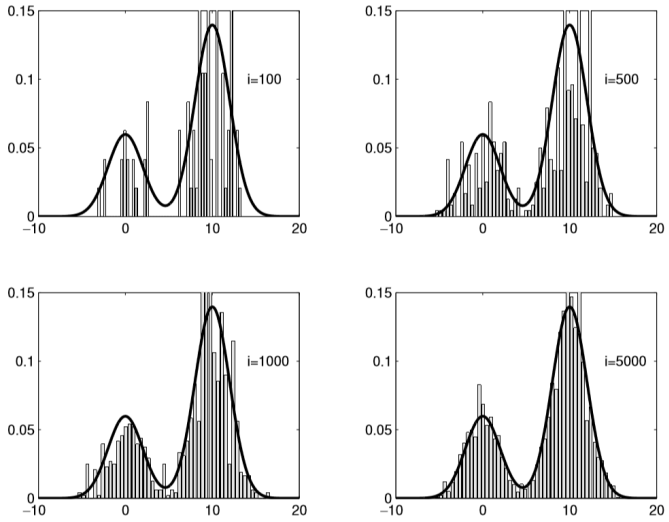

Figure 5: Sample distribution with Metropolis-Hastings algorithm MCMC. Solid line represents the target distribution and bars represent the sample distribution. This figure is taken from [7].

By recalling stationary distribution theorem, we see that Eq.19 is in the form $\pi = A\pi$ where $\pi = p(x)$. So our target distribution becomes the stationary distribution in Markov chain. Therefore the samples drawn from random walk will produce the $p(x)$ distribution. In fig.5 we see that the sampling distribution gets better compared to the target distribution with the increase of iterations in the random walk.

Although MH algorithm works very efficiently, we have to be careful in choosing the proposal distribution. As we can see in fig.6, the choice of Gaussian distribution for jumping probability can change the sampling distribution compared to the target distribution. If we choose too narrow Gaussian distribution (small σ) then the random walk will be trapped in a local maxima. If we choose too broad Gaussian distribution (large σ), random walk will be all over the sample region with high probability, thus missing peak distributions. Therefore we have to choose optimized proposal distribution so that we get fair sample distribution after significant number of iterations.

4.4 Application of MCMC

Now as we know how Markov chain Monte Carlo method works, we can compare the method with rejection sampling method. In rejection sampling method we discard all the samples which do not satisfy the condition thus wasting many sample points (i.e. wasting time in the sampling process). However in Markov chain Monte Carlo we never discard any of the samples. Even if we reject next sample (i.e. going to the next vertex in the graph) to draw, last sample will be selected for that iteration. Another difference with rejection sampling is that in rejection sampling it is very hard to draw sample from low probability region. On the other hand in MCMC method, jumping probability can take the random walk to low probability region

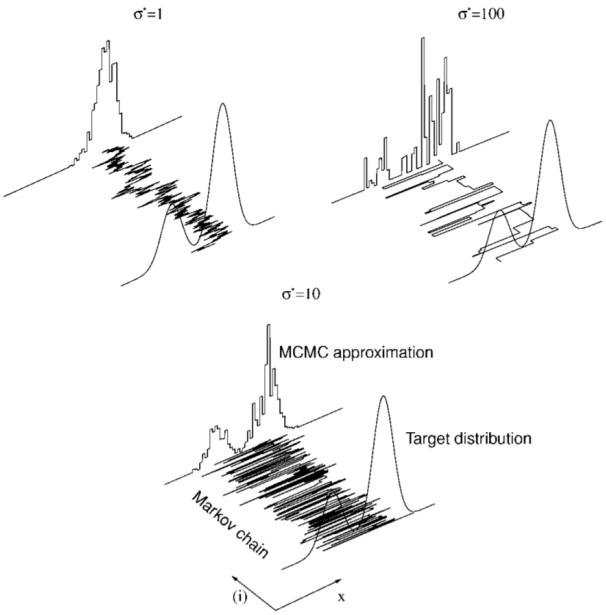

Figure 6: Representation of sample distribution (bar plot), target distribution (solid line) and markov chain (jumping tracks connecting two distribution). With different σ value of Gaussian jumping probability, sample distribution changes compared to target distribution. This figure is taken from [7].

and as we will mostly compare the random walk steps between low probability neighbours in that region, there is always a significant probability to draw sample for those low probability.

So we see that Markov chain Monte Carlo is very efficient for higher dimension numerical integrations, sampling from complicated probability distribution functions and optimization. We know that numerical integration, random sampling and optimization are the three basic building blocks for every simulation and statistical analysis in physics. Therefore Markov chain Monte Carlo has broad range of applications in almost every field of physics. Let us discuss some of the applications briefly in this section.

Bayesian Statistics:
In Bayesian statistics, statistical model is understood by describing the probability distribution of unknown parameters, $p(\xi)$ which is called 'prior distribution'. When we have some experimental data x we realize the probability distribution by expressing likelihood which is the distribution of observed data given the model parameters, $p(x|\xi)$. We then combine likelihood and prior distribution to update our understanding on the model parameters defined as posterior is expressed as,

$$p(\xi|x) = \frac{p(\xi)p(x|\xi)}{\int_\Theta p(\xi)p(x|\xi)}. \tag{20}$$

In this equation the normalising factor has to be computed numerically in order to find the posterior. MCMC method is used frequently in such computation.

Very often, objectives of some analysis are to obtain the expectation value of

some function f for variable x from some distribution where y is given with the expression,

$$E_{p(x|y)}(f(x)) = \int f(x)\ p(x|y)dx. \tag{21}$$

To find this expectation value, we also use Markov chain Monte Carlo method for numerical integration.

Statistical Mechanics:
In statistical mechanics, partition function Z of states s and Hamiltonian $E(s)$ is expressed as,

$$Z = \sum_s exp\left(-\frac{E(s)}{kT}\right) \tag{22}$$

where k is the Boltzmann's constant and T is the temperature of the system. We know calculation of partition function is important in statistical mechanics to understand the microstates and thermodynamic properties of the system. However analytic calculation of each state s and summing them to find the partition function is time expensive. So we use MCMC to calculate such partition function efficiently.

Signal processing:
In signal processing autoregressive (AR) model describes several time varying properties of the signal which is used for several applications e.g. filtering (low pass, high pass, band pass filters etc). AR models are classes of different random processes. Markov chain Monte Carlo is used to simulate such models in order to understand their behaviour in a signal processing unit. There are several modes of AR models, for example, zeroth model AR(0) is the model for white noise, AR(1) represents red noise. Therefore by simulating such models through MCMC method we can understand the noise in our electronics and by comparing simulated samples with actual noise data we can find the associated parameter values by performing statistical analysis.

Optimization The use of MCMC method in optimization is also important in many fields of physics. If we want to solve a problem with exact solution, we might spend long time n the computation because of dimensionality and complexity of the problem. In such cases, optimizing the solution as close to the exact solution as possible can be very efficient. We can use MCMC method to find optimized solution by minimizing some objective function from large set of possible solutions.

5 Conclusion

Likelihood ratio test and p-value calculation are well established analysis tools for experimental results. However in realistic scenarios, where assumptions for statistical theorems are not followed in experimental observables, we take advantage of the use of Monte Carlo methods. In my report, I have discussed how importance sampling and rejection sampling Monte Carlo work and their inefficiency for complex systems. Then we have moved to the discussion of MCMC method and its efficient application any many fields. In summary, Monte Carlo methods and their use in

every field of physics (condensed matter physics, particle physics, astrophysics and cosmology and many more) have become an integrated part of research components. Hopefully this project report will help the readers to get some understanding on the overview of statistical analysis and working principle and application of Monte Carlo methods as one of the general components of experimental methods.

References

[1] Cern data centre passes 100 petabytes, Mar 2015.

[2] George Casella and Roger L. Berger. *Statistical Inference*. 2001.

[3] K Nakamura and Particle Data Group. Review of particle physics. *Journal of Physics G: Nuclear and Particle Physics*, 37(7A):075021, 2010.

[4] p-value figure, wikipedia. `https://en.wikipedia.org/wiki/P-value#/media/File:P-value_in_statistical_significance_testing.svg`. Accessed: 2017-04-13.

[5] S. S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Ann. Math. Statist.*, 9(1):60–62, 03 1938.

[6] M. Wiebusch. Numerical computation of -values with myfitter. *Computer Physics Communications*, 184(11):2438 – 2445, 2013.

[7] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1):5–43, 2003.

[8] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.