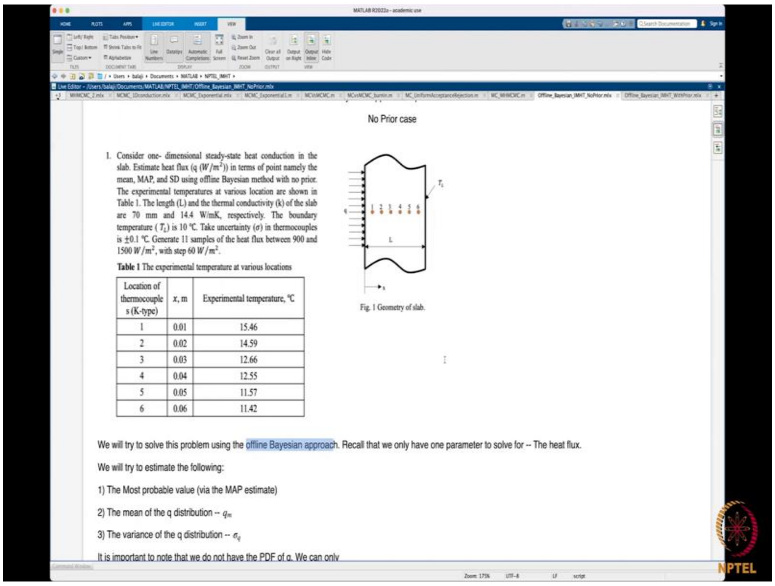
Inverse Methods in Heat Transfer
Prof. Balaji Srinivasan
Department of Mechanical Engineering
Indian Institute of Technology - Madras

Lecture - 41
MHMCMC for Inverse Problems

(Refer Slide Time: 00:19)



No Prior case

1. Consider one- dimensional steady-state heat conduction in the slab. Estimate heat flux (q (W/m^2)) in terms of point namely the mean, MAP, and SD using offline Bayesian method with no prior. The experimental temperatures at various location are shown in Table 1. The length (L) and the thermal conductivity (k) of the slab are 70 mm and 14.4 W/mK, respectively. The boundary temperature (T_L) is 10 °C. Take uncertainty (σ) in thermocouples is ± 0.1 °C. Generate 11 samples of the heat flux between 900 and 1500 W/m^2, with step 60 W/m^2.

Table 1 The experimental temperature at various locations

Location of thermocouples (K-type)	x, m	Experimental temperature, °C
1	0.01	15.46
2	0.02	14.59
3	0.03	12.66
4	0.04	12.55
5	0.05	11.57
6	0.06	11.42

Fig. 1 Geometry of slab.

We will try to solve this problem using the offline Bayesian approach. Recall that we only have one parameter to solve for -- The heat flux.

We will try to estimate the following:

1) The Most probable value (via the MAP estimate)

2) The mean of the q distribution -- q_m

3) The variance of the q distribution -- σ_q

It is important to note that we do not have the PDF of q. We can only

Welcome back, this is week 7 of inverse methods in heat transfer and what I wish to cover within this video is this algorithm called the Metropolis Hastings Markov Chain Monte Carlo algorithm. It is a long name; it is also called MHMCMC Metropolis Hastings Markov chain Monte Carlo. Now each of these parts does something in the kind of problems that we are solving. Now what I want to do here before jumping into the algorithm is to discuss the motivation for why we are doing this.

Now some of it was implicit in what I discussed so far. Now remember what we are doing we are trying to solve for some parameters. we are finding some parameters some w. Now this w as we have seen earlier could consist of a large number of parameters, let us say n parameters. Now typically here we have just two parameters or in the problems that we were solving this week at least we had only one parameter.

Now the question always is this we want the PDF or the joint PDF of this w_1 through w_n these n parameters that is what range does w_1 vary in what range does w_2 vary. So, on and. So, forth till w_n. Now in order to do that let us say we have just one parameter like we were doing before in the previous video suppose only one parameter q then what we want basically is the probability distribution function of q. It could look like something like this even though we were looking like looking at Gaussians.

Now imagine we have some such distribution but we can sample our constraint is that we can sample only at finite points. Why is that? Because remember for each value of q we actually had to calculate what the model value of T was and then find out $(\hat{T}-T)^2$etc and that is really the only way in which we could generate this value of the PDF of q at one point. So, just each sample is expensive, it is not cheap to sample at each point.

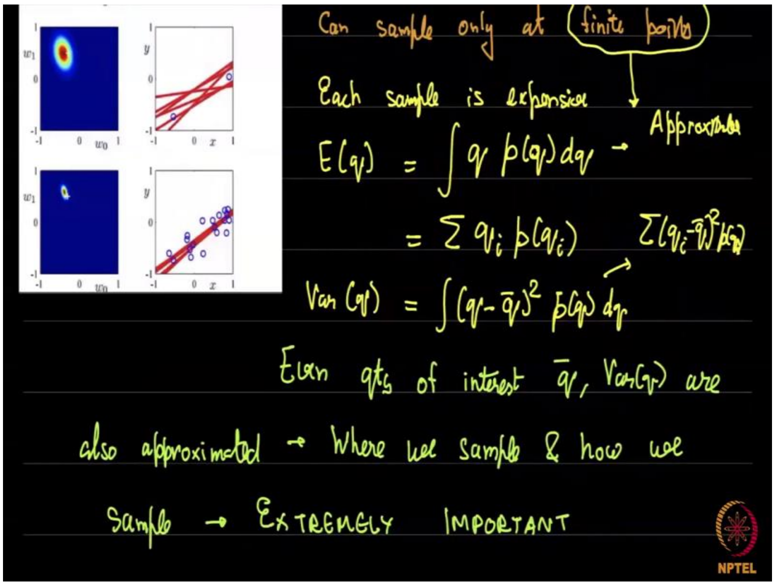
Now this is for a simple slab imagine if it is like a full 3D simulation where you have to calculate at each point and you have lots of thermocouples then each sample becomes really expensive. So, the first point is we can sample only at finite points, the second point is that when we calculate something like $E(q)$. $E(q)$ is $\int q$ at a particular point multiplied by let us say the PDF of $q\ is\ p$.

Even this integral can only be approximated. how are we approximating it? we are approximating it using $\sum q_i\,P(q_i)$. So, what we do effectively is we take a few points, let us say we call this q_i and we sum just at this point. And we take a small interval around this that is basically what we are doing this works out because the integral of q_i turns out to be one and that is why this integral basically equates to the summation because of normalization.

Only for variance you had to do $\sum(q-\bar{q})$ or $E(q)p(q)dq$ and this was also approximated by $\sum(q_i-\bar{q_i})^2P(q_i)$. Now what is the problem? So, there are two problems one is first we are sampling only at finite points and even the quantities of interests that is what is the mean value of the parameter, what is the variance of that parameter or also approximated.

So, these are key problems that we are dealing with. Now because of this where we sample and how we sample becomes extremely important.

(Refer Slide Time: 05:43)



Can sample only at finite points

Each sample is expensive

Approximate

$$E(q) = \int q\, p(q)\, dq \rightarrow$$

$$= \sum q_i\, p(q_i) \qquad \sum (q_i - \bar{q})^2 p(q_i)$$

$$Var(q) = \int (q - \bar{q})^2 p(q)\, dq$$

Even qts of interest $\bar{q}$, $Var(q)$ are also approximated → Where we sample & how we sample → EXTREMELY IMPORTANT

Now in this Metropolis Hastings Markov chain Monte Carlo there are three parts, the first part you actually the causation is like this, first part is Monte Carlo, from there we go to Markov chain and from there we go to Metropolis Hastings. Now what is the basic essence of Monte Carlo the basic essence of Monte Carlo is instead of doing continuous sampling or sampling the entire sample space you just use some sample points.

So, this is a subset of sample points from the sample space, if you take this that is basically the idea of harmonic order. Remember we add our entire sample space whether it is continuous or even if it is discrete in fact the initial way in which Monte Carlo was done was for a discrete random variable but nonetheless even if it is a large sample space out of that we sample from a few places.

Now when we do offline sampling, these are fixed and these are independent samples. All samples are independent if you do offline sampling or offline Bayesian but how do we know that we are sampling at the right place. for example, let us deal with if this is was this was something like our actual PDF and this is our sample space. let us say this is a full sample space let us say minus one to one that is not accurate but let us say that is the full sample space.

Now only some portion of it actually contributes to the integral, but somehow let us say when you were sampling you only sample it at these places. you will actually end up getting something like $\bar{q}$ or $\overline{w_0}$ equal to zero because you really have nothing going on there or very little go will go on. So, if you do not sample from this portion which is the important portion, you will actually under sample or you will very poorly sample.

Now when we do pure Monte Carlo or we do pure offline we are basically assuming that I will just put a lot of points here and somehow it will magically capture it. Now come if you come to something like this, it is possible that you might have put samples of this sort and you will get practically nothing of this PDF. if you just sample here you get practically nothing which is when we come to this idea of both Markov chain as well as Metropolis Hastings.

So, the idea is this, the idea is in Markov chain, well the general idea of a Markov chain is any sample let us say w_{t+1} is just dependent on w_t. Another way to say it is the new sample depends only on the previous sample, on just the previous sample. Now imagine instead of doing a grid search like, this you start doing a different grid search. The grid search works like this you first give one point.

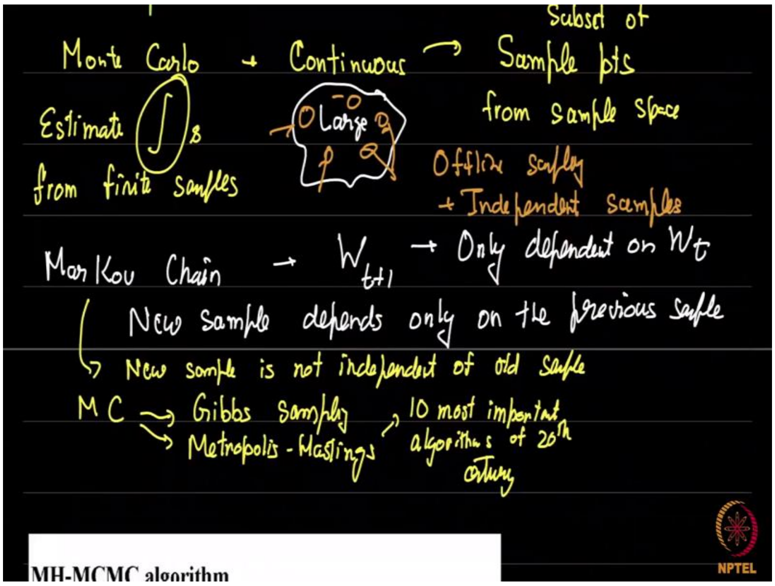
You look at the probability here and you are not very happy, looks like I am in a low probability space. I will randomly move somewhere else and then you look around and somehow you have a good guess for where to go next. So, this is somewhat like a random walk, I will shall tell you how to do this better. Now what you do is? you do a lot of these random walks earlier versions of Marco chain Monte Carlo were that you start from various places randomly sample the domain at various places, but the new sample depends on just the previous sample.

So, you search a little bit more regularly than what I am saying or in a more organized fashion than what I am saying but nonetheless it is not a simple grid search. So, this is a unique type of sampling just Marco chain, Marco chain has multiple types of sampling. So, Marco Chain simply says that new sample is not independent of old sample. Now within Marco chain there are multiple types of samples, one such type of sampling is called Gibbs sampling, another is called Metropolis Hastings which is what we will do.

Now this Metropolis Hastings sample has been named as one of the most important algorithms. So, 10 most important algorithms of the 20th century, another one is the fast Fourier transforming. This is very important in multiple fields in economics, it is important in biomedicine, it is important of course an inverse heat transfer it is important. So, a large number of places where we have to estimate, so, this is true in general of Monte Carlo.

We have to estimate integrals from finite samples and we do not know you know what is the density of that integral, where is it high, where is it low, when we do not know such information but you still want to evaluate integrals of the sort that we have in expectation or in variance or in the denominator in such cases Monte Carlo method and specifically not Monte Carlo but the Markov Chain Monte Carlo method becomes important Metropolis Hastings is a certain type of sampling here.

(Refer Slide Time: 12:34)



Monte Carlo + Continuous → Subset of Sample pts from sample space

Estimate $\int$s from finite samples

Offline sampling + Independent samples

MarKov Chain → W_{t+1} → Only dependent on W_t

New sample depends only on the previous sample

→ New sample is not independent of old sample

M C → Gibbs sampling
→ Metropolis - Hastings

→ 10 most important algorithms of 20th century

MH-MCMC algorithm

NPTEL

So, now before I am going to describe the algorithm here but let me give you an intuition. So, for how this happens. So, let us again go back to the simple probability case, we were looking at which was we were sampling from q. So, let us say the PDF of q looked somewhat like this. So, this is PDF of q and the way we sampled it is, we start at some random Point within the q domain. So, let us say this random point was here.

So, we start with this random point and you do your sampling basically, you find out T simulation etcetera and find out what its value is and indeed you find out PDF at this point. So, let us call this PDF one. Now you have a choice on where to take the next point. So, this is a Markov chain. The Marco chain will say that at this point I will look somewhere in this neighbourhood and I will look somewhere in the neighbourhood and a lot of points in this neighbourhood that is somewhere here, this is I am going to search in this proposal region.

That is, I will not randomly jump from here to let us say somewhere here like I started here I don't want to jump somewhere very far off, but I will say I will search once again with a certain probability. So, let us say I search with the probability which is a Gaussian. So, this is called the proposal distribution. So, what is the idea you choose an initial point the next point that you choose is chosen with a certain probability in the neighbourhood of the current point.

So, this can be any distribution but we will use as usual we'll use a Gaussian, but the Metropolis Hastings algorithm works regardless of what distribution you put here with of course a certain constraints on what type of probability, that you are going to use, you could use a uniform distribution, that is all points in this neighbourhood are equally likely or you can say that I will put most probable choice of next point is right here and you can choose a little bit to the left and a little bit to the right etcetera.

But the idea is the next point. So, the next point q new is some probability centered on q old with let us say since I am assuming it is a normal probability with some Sigma. So, Sigma let us call it Sigma s for Sigma for searching. So, let us say this new Point turned out to be here.

$$q_{new} \sim \aleph(q_{old}, \sigma_s)$$

This is my new point; it does not mean I always have to sit within the Gaussian occasionally I can come really far with the Gaussian means Gaussian of course has infinite width to the left and right.

So, now once again you calculate what the probability is? now once you calculate the probability you notice that again this point has higher probability than my previous point. So, that is $pdf(q_{new}) > pdf(q_{old})$. When you see this, you have a choice on whether to accept q new amongst the sample set or not. what are we doing when we do this. So, ultimately what we are doing is remember when we are putting sample points let me come here.

Our original choice when we were using offline Bayesian was a set of random points or not a random point an organized set of points like this, in offline Bayesian. This is what decides our new sample. Now instead of using this queue sample you are proposing a new q sample where you started somewhere, the next point is a proposed point. Now you choose to keep this point or not if you chose to erase this point, you will choose for a better point.

Now what is our aim while doing this we want to somehow random. So, that most of the proposal points are here because that is where most of the integral is coming from. So, ideally our proposal points should look like this. So, let us say I mark it in pink, lot of points here and a few points here. So, what we want is our proposed points or not our purpose points, finally our sample points should be such that a lot of points are in the high PDF region and very few points are in the low PDF region.

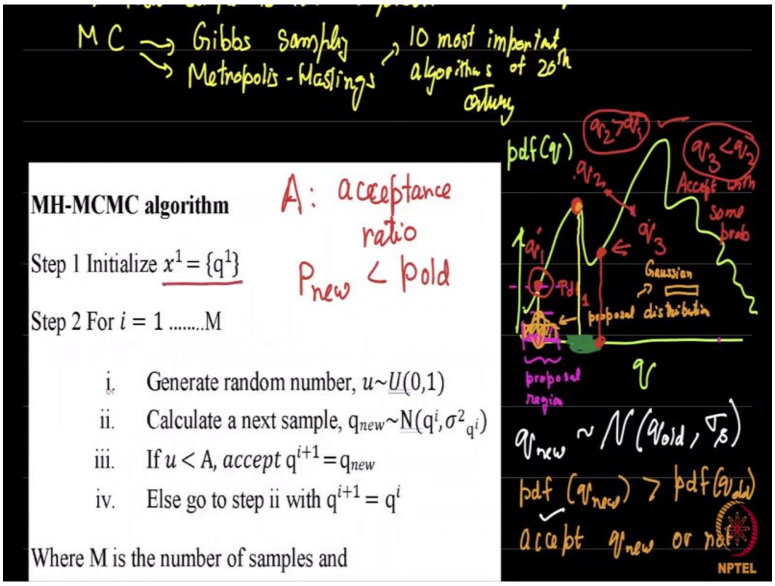
So, similarly when we come here and we propose new points, we want to decide new points based on whether they are better compared to the whole point or not. Now what is better if my probability is high obviously this is a better Point than the previous one. So, I will say, that if PDF is of q new is greater than PDF of q old, we will accept it. Now what happens if you gave a proposal point, let us say your old proposal point was or your old point was this or you accepted this and you again made a region.

So, let us say I kept a region here and started searching and the next proposed point was here. Now this proposal let us call this q_3. So, this is q_2, q_1. q_2 was greater than q_1. So, I accepted it. Now I came to q_3. Now remember this is Markov chain I am not going to look q_3 with respect to q_1, I can't look at q_1, I can only look at q_2. Now q_3 I will check with respect to q_2. Now q_3 is worse than q_2 and I do not remember what q_1 was because it is Markov chain, I have forgotten it. Now q_3 less than q_2 should I take it or not.

This is the basic question. Now what Metropolis history says is accept with some probability. So, if it is greater always accept if it is lesser accept sometimes and reject sometimes. Now the cleverness of metropolis Hastings is the ratio with which it accepts A is called the acceptance ratio this acceptance ratio happens, when probability of the new point is lesser than the probability of the old point you will accept it with some ratio and reject it with some ratio.

So, let us now look at what the algorithm is? the entire algorithm is simple and it will become even clearer when I actually show you a simulation of exactly the same slab case with a metropolis Hastings Marco chain Monte Carlo algorithm. So, let us go through the algorithm now. So, the idea is like this you first initialize and make some initial guess like I said you randomly choose a certain point within the domain where you choose to initialize.

(Refer Slide Time: 20:48)



MH-MCMC algorithm

Step 1 Initialize $x^1 = \{q^1\}$

Step 2 For $i = 1$ ....M

```
i.    Generate random number, $u \sim U(0,1)$
ii.   Calculate a next sample, $q_{new} \sim N(q^i, \sigma^2_{q^i})$
iii.  If $u < A$, accept $q^{i+1} = q_{new}$
iv.   Else go to step ii with $q^{i+1} = q^i$
```

Where M is the number of samples and

So, I will erase this here and draw this once more. So, that the process becomes a little bit clear on how we are doing this. So, let us say I have a probability distribution like this and this is my q, where I am searching, I am again showing for a single parameter, of course, this works for multiple parameters also. So, suppose I initially guessed somewhere here this is my q_1. Now once I choose q_1, I Now have to decide on q_2. So, first I find q_1 at q_1 I find out P_1.

Now for q_2 I propose a distribution here. Now this proposed distribution comes at this step, it is centered it is a normal distribution, it is centered around the previous distribution. So, q_2 is a random number in this range, it is highly probable that it will choose q_1 itself, of course that will never happen in practice computational. But the closer it is to q_1 the more likely it is to get selected and the farther away, it is from q_1 the less likely it is to get selected.

So, you go here and let us say at some point you select few this q_2 gives you a certain probability P_2. Now that is what is written here but not so, clearly, all it says is you generate a random number U, this is a uniform random variable between 0 and 1. Now why do we do that because we are trying for this very simple idea that if P_2 greater than P_1 accept, if P_2 less than P_1 accept with probability A.

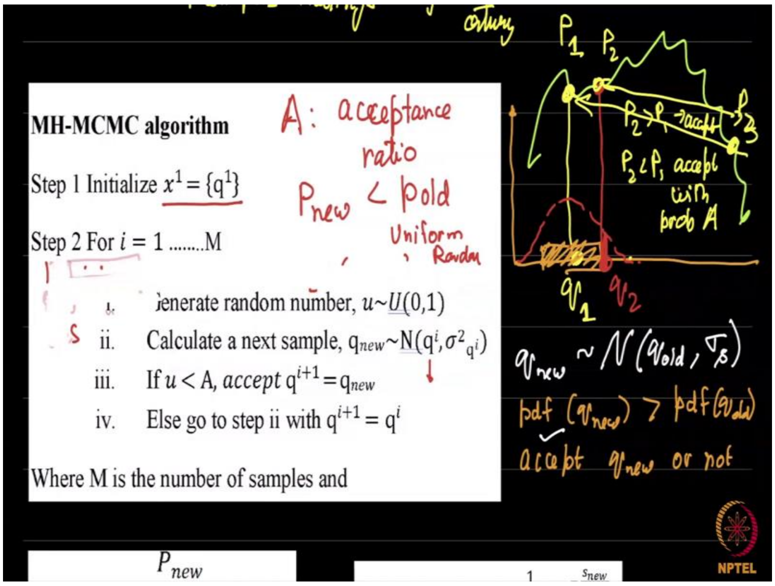
Now how do we do this let us say this probability A turns out to be 0.5. So, let us say we went to these two points and it actually turned out that P_2 was some point here let us say. So, farther off and P_2 is less than P_1 for let us say this is P_3 and P_3 is less than P_2 that is where we came and we want to accept it with probability 0.5. Why 0.5, I am making something up how would we achieve that in computational practice what you do is this you select a number between 0 and 1 this step.

Now it is a uniform probability, which means you could have selected any number between 0 and 1 with equal probability and A here is 0.5. So, if the number that you randomly selected turned out to be less than 0.5, you will reject it, if it turned out to be greater than 0.5 you will accept it. what does this mean? in 50 percent of the chain cases you will reject it and 50 of the cases you will accept it because you have put A exactly at the place where you want the probability.

So, if you want to accept with probability A all that means is you put a line, you choose a random number in the range 0 to 1 and let us say a was 0.25, you say well if it is less than 0.25, I will accept it, if it is greater than 0.25, I will reject it, what that will do is it will collect it will select it only in 25 percent of the case. So, that is the basic trick that we are applying here we use a random number uh uniform random number in order to sample from case which will give us exactly an acceptance ratio of A.

Now what is this A? What is this magical number A again that is what I will show next.

(Refer Slide Time: 25:14)



MH-MCMC algorithm

Step 1 Initialize $x^1 = \{q^1\}$

Step 2 For $i = 1$ ....M

```
    i.   Generate random number, $u \sim U(0,1)$
    ii.  Calculate a next sample, $q_{new} \sim N(q^i, \sigma^2_{q^i})$
    iii. If $u < A$, accept $q^{i+1} = q_{new}$
    iv.  Else go to step ii with $q^{i+1} = q^i$
```

Where M is the number of samples and

A happens to be now notice,

$$A = \min\left(1, \frac{P_{new}}{P_{old}}\right)$$

what does this mean? let us first take two cases. So, $A = \min\,(1, \frac{P_{new}}{P_{old}})$, P_{new} represents the new probability and P_{old} represents the old probability. Let us say $P_{new} > P_{old}$ and this would mean that $\frac{P_{new}}{P_{old}} = r > 1$. So,

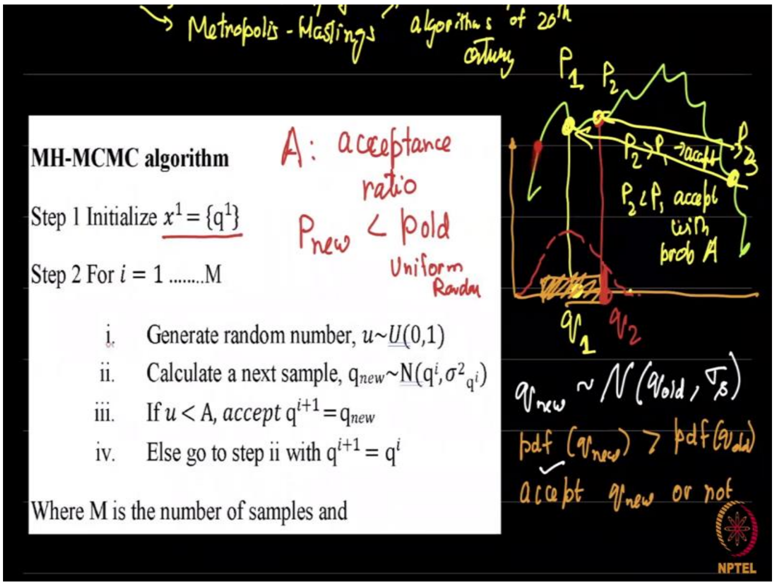
$$A = \min(1, r)\,; r > 1$$
$$A = 1$$

Which means that A equal to 1 means always accept because we accept it with probability 1. For example, if I come here, I will generate a uniform random number. this uniform random number will be between 0 and 1 and what is A, A is one. So, this number will be always even at the worst-case scenario will be 0.999, this will be always less than 1. So, you will always accept this new point this basically means that new points with $P_{new} > P_{old}$ are always accept.

But let us say $P_{new} < P_{old}$. So, let us say $P_{new} < P_{old}$ and the value is $\frac{P_{new}}{P_{old}}$ is let us say 0.5 this would mean A is 0.5 which means accept 50 percent of the times. So, if I draw this probability distribution, let us say P new turned out to be somewhere here really lower compared to P2, let us say it is 10 times lower than P2 then it will be accepted only 10 percent of the times if it is a million times smaller than P2.

Then 1 million chances that you will accept this really bad point by why do you want to accept the bad points because it is unclear what this looks like.

(Refer Slide Time: 27:48)



MH-MCMC algorithm

```
Step 1 Initialize $x^1 = \{q^1\}$
Step 2 For $i = 1$ ........M
    i.   Generate random number, $u \sim U(0,1)$
    ii.  Calculate a next sample, $q_{new} \sim N(q^i, \sigma^2{}_{q^i})$
    iii. If $u < A$, accept $q^{i+1} = q_{new}$
    iv.  Else go to step ii with $q^{i+1} = q^i$
Where M is the number of samples and
```

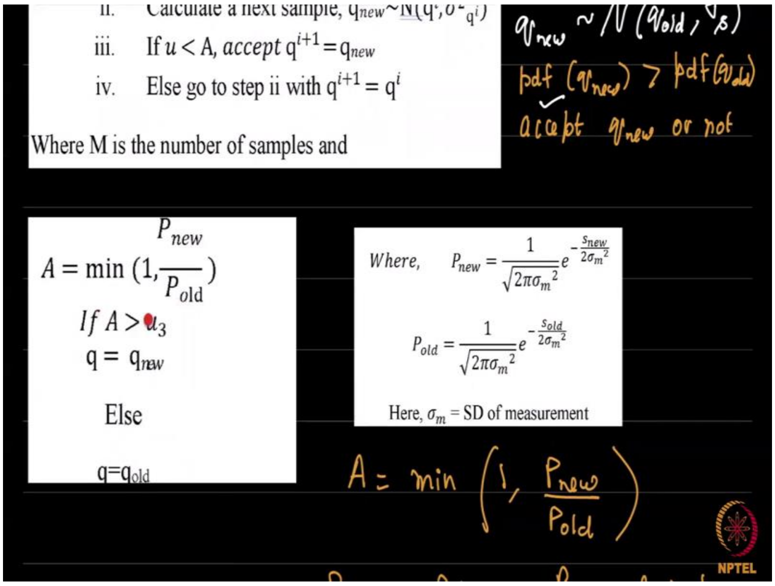
Metropolis - Hastings algorithms of 20th century
A: acceptance ratio
$P_{new} < P_{old}$
Uniform Random
P_1
P_2
P_3
$P_2 > P_1 \to$ accept
$P_2 < P_1$ accept with prob A
q_1
q_2
$q_{new} \sim N(q_{old}, \sigma_s)$
pdf $(q_{new}) >$ pdf (q_{old})
accept q_{new} or not
NPTEL

It is possible that our distribution function could look really weird like and suddenly it may jump up. So, even though this value is really low in comparison to this nearby another region is there which is important. So, in order to account for that we actually start accepting points which are not necessarily good points immediately but they might still turn out to be decent points. Now what is P new and P old this we have already seen before.

P new is just the probability of the new points remember for our slab example or any linear regression example it is simply summation of the error square and similarly P old is summation of the old error square. So, we basically calculate the PDF of this new point and old point and accept new points always if they are better than just the previous point. But reject new points in case they are or accept them with a certain probability in case they are a little bit worse.

So that is what is written here in pseudo code this is if A is greater than that random number you take it otherwise you reject it. So, I will come to this the code for this but let me just show you what would happen in case we did it in 2D, in the same idea if we did this in 2D what it would look like. So, what we expect it to look like is this we start at some random point, we start here let us say and it starts proposing.

(Refer Slide Time: 29:23)



```
ii.  Calculate a next sample, $q_{new} \sim N(q^i, \sigma^2_{q^i})$
iii. If $u < A$, accept $q^{i+1} = q_{new}$
iv.  Else go to step ii with $q^{i+1} = q^i$
```

Where M is the number of samples and

$q_{new} \sim N(q_{old}, \sigma_s)$

$pdf(q_{new}) > pdf(q_{old})$ ✓

accept q_{new} or not

$$A = \min\left(1, \frac{P_{new}}{P_{old}}\right)$$

$If\ A > u_3$

$q = q_{new}$

Else

$q = q_{old}$

Where, $P_{new} = \frac{1}{\sqrt{2\pi\sigma_m{}^2}} e^{-\frac{S_{new}}{2\sigma_m{}^2}}$

$P_{old} = \frac{1}{\sqrt{2\pi\sigma_m{}^2}} e^{-\frac{S_{old}}{2\sigma_m{}^2}}$

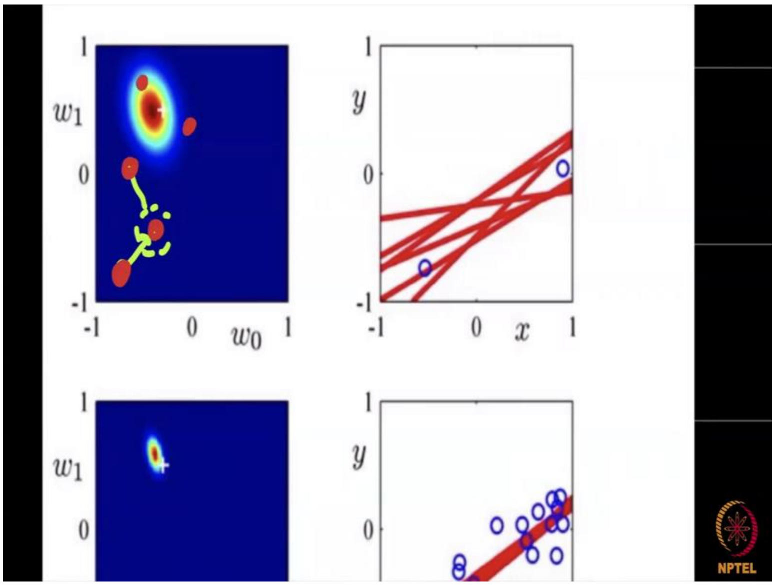
Here, σ_m = SD of measurement

$$A = \min\left(1, \frac{P_{new}}{P_{old}}\right)$$

Let us say one point I put a few points here but let us say proposes a point here you see if somehow magically this probability was increasing this will accept it. It proposes another point maybe it is slower you reject it, but it proposes a point here and this is accepted. So, this one is now erased. Now this point is accepted. So, let us put red wherever we accept the point. So, this one is accepted, this one is accepted, this one is accepted, let us say this one is accepted this one is accepted.

And once you come here you see this is a high probability region and you will be unlikely to go out you will start accepting points here and you will start getting a lot of points here and it is unlikely that you will step up.

(Refer Slide Time: 30:15)



Now typically what happens is we have something called a burn in Period. the burn -in period is that the first B let us call this build the first B samples are not used for integration. So, if you are summing up let us say q e to the power minus s by 2 Sigma square instead of doing it from 1 to the total number of samples you will actually do it from B or B plus 1 to the total number of samples. Because the first B samples will typically be in this bad region there are other reasons also, I am just giving you a casual reason.

So, they might be actually in a bad region and your real samples are somewhere here. So, what is this B this is as we saw in general machine learning also this is what is a hyper parameter you have to select what this B is. But the idea is very simple you cycle over a few points NS points reject the first be and or throw away burn the first B select the rest for integration. Now when I say cycle, within the cycle of course there is accept it. it is after this entire cycle is over the Metropolis Hasting cycle is when you start doing the burning.

Now this Metropolis Hastings lies in the way we declare what A is and how we generated The Proposal region. Now if you have different ways of generating. Now in general not only normal you can have other distributions it is still called metropolysis in such a case but how we use the acceptance ratio Etc decides because the other one is like I told you Gibbs sampling. Now a major advantage of Metropolis Hastings is it is what is called sample efficient.

As a number of parameters increases, for example, pure Monte Carlo with basically offline might be good for the type of examples that we did for low parameters. So, when you have to partition essentially a one-dimensional domain and you are just selecting here Markov simple Monte Carlo might be efficient but once you come to 2D it becomes less efficient when you come to 3D it becomes even lower efficient.

So, its efficiency starts dropping whereas Metropolis Hastings starts working better and better. Especially in finance Industries or in fact even in engineering problems if you have a few thousand parameters, then really pure Monte Carlo will not work well at all. you will have to use something like Metropolis Hastings in order to do this. So, just to recap what we did is that we saw that just Monte Carlo uses some systematic prior sample.

Whereas the Markov chain tells you that the next sample will depend on the previous sample and Metropolis Hastings has a specific way of selecting the next sample it accepts a few and rejects a few, at the end of it you have a bunch of sample points that are usually efficient in concentrating on the regions of the probability space where most of the probability is actually concentrated.

So, that is the basic idea behind Metropolis Hastings Markov chain Monte Carlo. In the next video I will go back to the slab example and show you with the same slab example I will show you what happens if we do this exactly the same algorithm that I showed you just now. So, I will see in the next video where you will see an interesting demonstration, thank you.