STOCHASTIC APPROXIMATION: THEORY AND APPLICATIONS

Dr. Gugan Thope

Department of Computer Science and Automation

Indian Institute of Science

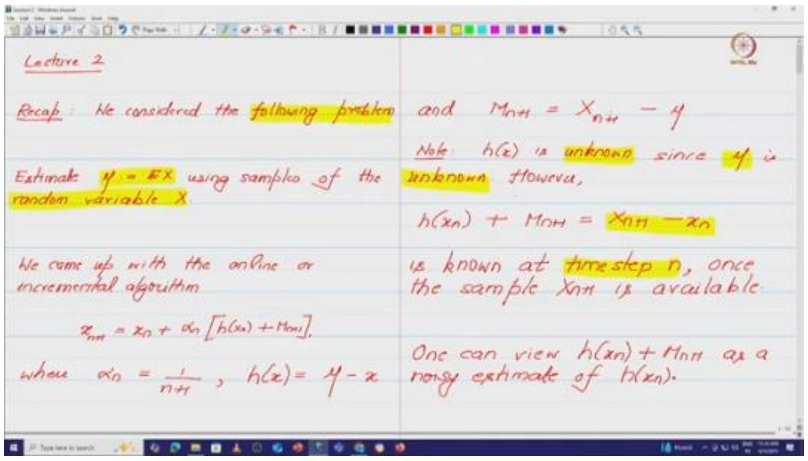
Lecture 2

Estimating the Mean of a Random Variable

Hello and Namaste everyone. Welcome to Lecture 2 of this course on Stochastic Approximation. Let us first do a quick recap of what we did in the previous class. In the previous class, we looked at the following problem, which involved estimating the mean of a random variable X using independent and identically distributed samples of the random variable X itself. We then came up with the online or incremental algorithm: $X_{n+1} = X_n + \alpha_n (h(X_n) + M_{n+1})$, where $\alpha_n = 1/(n+1)$, and we refer to α_n as the step size.

$h(X) = \mu - X$, and we refer to this h function as the ideal update direction. $M_{n+1} = X_{n+1} - \mu$, and we refer to M_{n+1} as the noise in the estimate of $h(X_n)$. So, clearly, $h(X)$ is unknown. It is unknown because μ is unknown. Recall that our goal is to actually estimate μ. So, of course, μ is unknown, and hence $h(X)$ is unknown for any X.

However, if you add $h(X_n)$ and M_{n+1}, we get the quantity $X_{n+1} - X_n$. $X_{n+1} - X_n$, and this quantity is indeed known at time step n once the sample X_{n+1} is available. So, let me again summarize. We do not know $h(X_n)$ because μ is unknown. However, we have access to the noisy estimate of $h(X_n)$, which is $h(X_n) + M_{n+1}$, and this noisy estimate becomes available when X_{n+1}, which is the (n+1)th sample of the random variable X, becomes available.



Lecture 2

Recap: We considered the following problem

Estimate $\mu = \mathbb{E}X$ using samples of the random variable X.

We come up with the online or incremental algorithm

$$x_{n+1} = x_n + \alpha_n [h(x_n) + M_{n+1}].$$

where $\alpha_n = \frac{1}{n+1}$, $h(x) = \mu - x$

and $M_{n+1} = X_{n+1} - \mu$

Note: $h(x)$ is unknown since μ is unknown. However,

$$h(x_n) + M_{n+1} = X_{n+1} - x_n$$

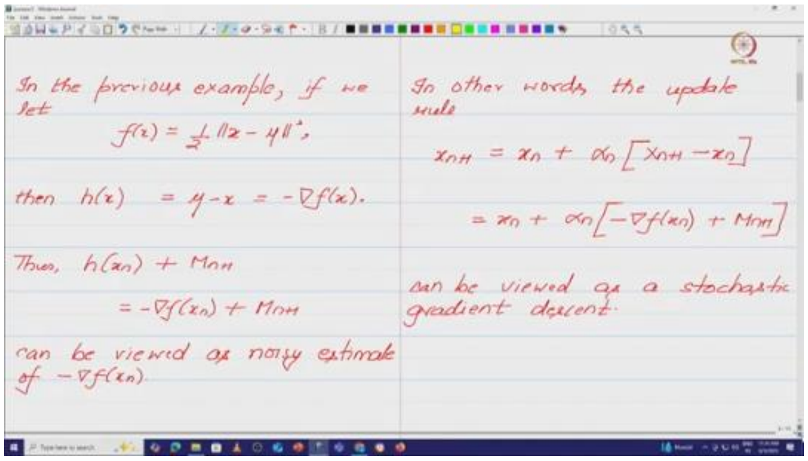
is known at timestep n, once the sample X_{n+1} is available.

One can view $h(x_n) + M_{n+1}$ as a noisy estimate of $h(x_n)$.

Right? So, now let us see if we can come up with a better interpretation of what this algorithm is trying to do. Right? So let us define a function f given by f of x equals half of norm of x minus mu the whole squared. Right?

$$f(x) = \frac{1}{2}||x - \mu||^2,$$

$$X \in \mathbb{R}^d, f: \mathbb{R}^d \to \mathbb{R}$$



In the previous example, if we let

$$f(x) = \frac{1}{2}\|x - \mu\|^2,$$

then $h(x) = \mu - x = -\nabla f(x)$.

Thus, $h(x_n) + M_{n+1}$

$$= -\nabla f(x_n) + M_{n+1}$$

can be viewed as noisy estimate of $-\nabla f(x_n)$.

In other words, the update rule

$$x_{n+1} = x_n + \alpha_n [X_{n+1} - x_n]$$

$$= x_n + \alpha_n [-\nabla f(x_n) + M_{n+1}]$$

can be viewed as a stochastic gradient descent.

So what we are imagining is that x is a d-dimensional random variable and f is a map from Rd to R which is given by this update rule over here, right? Now, in the previous slide, we saw that h of x equals mu minus x, and this expression can be viewed as the negative of the gradient of this function f, which is given above, right? h of xn plus mn plus 1 can be viewed as the negative of the gradient of f of xn plus mn plus 1.

$$h(x) = \mu - x = -\nabla f(x)$$

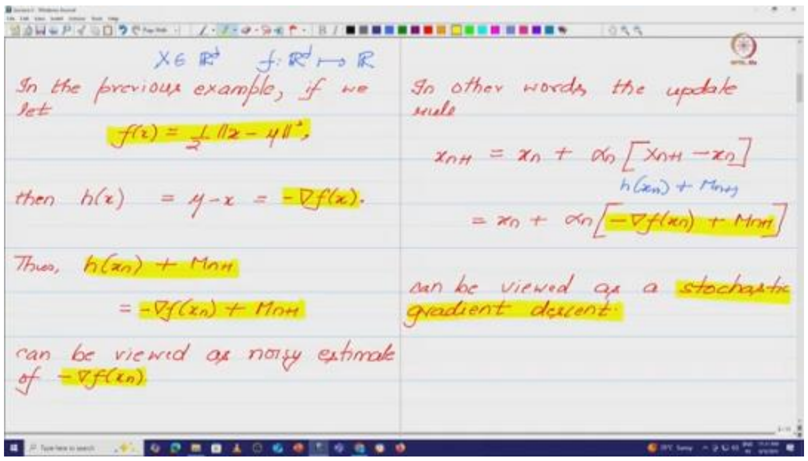
$$h(x_n) + M_{n+1} = -\nabla f(x_n) + M_{n+1}$$

In other words, this noisy estimate of h of xn is actually a noisy estimate of the negative of the gradient of this function f itself. In other words, the update rule xn plus 1 equals xn plus alpha n capital xn plus 1 minus xn, which was this h of xn plus mn plus 1, actually equals minus grad f of xn plus mn plus 1.

$$x_{n+1} = x_n + \alpha_n[X_{n+1} - x_n]h(x_n) + M_{n+1}$$

$$= x_n + \alpha_n[-\nabla f(x_n) + M_{n+1}]$$

Thus, this update rule can actually be viewed as a stochastic gradient descent with regard to this function f. We say stochastic gradient descent because this negative of the gradient is not available, as mu is unknown. Instead, we have access to minus grad fxn plus mn plus 1, which can be viewed. Excuse me, which can be viewed as a noisy estimate of this negative gradient. Hence, this update rule can be viewed as a stochastic gradient descent method, and if you recall my discussion from the first class, stochastic gradient descent is actually a special case of a stochastic approximation algorithm.



$X \in \mathbb{R}^d \quad f: \mathbb{R}^d \mapsto \mathbb{R}$

In the previous example, if we let

$$f(x) = \frac{1}{2}\|x - y\|^2,$$

then $h(x) = y - x = -\nabla f(x)$.

Thus, $h(x_n) + M_{n+1}$

$$= -\nabla f(x_n) + M_{n+1}$$

can be viewed as noisy estimate of $-\nabla f(x_n)$.

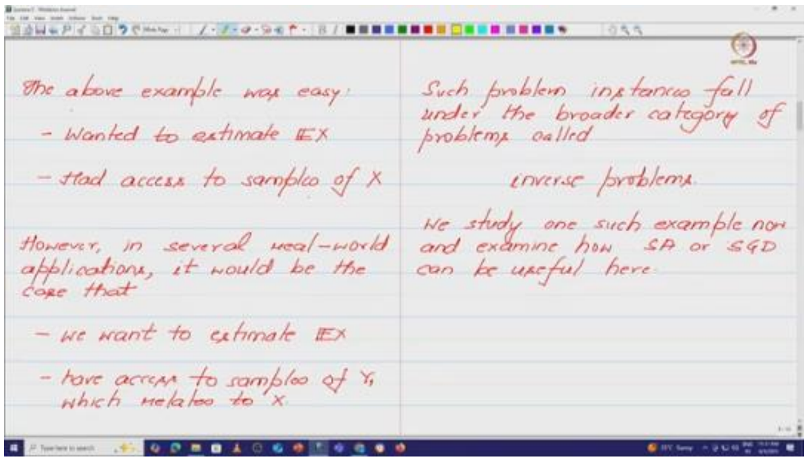
In other words, the update rule

$$x_{n+1} = x_n + \alpha_n \underbrace{[X_{n+1} - x_n]}_{h(x_n) + M_{n+1}}$$

$$= x_n + \alpha_n[-\nabla f(x_n) + M_{n+1}]$$

can be viewed as a stochastic gradient descent.

Alright, so in a sense, the above example, which involved estimating the mean of X, was relatively easy because of two reasons. We wanted to estimate the expected value of X. Okay, so there is a random vector X, and we wanted to estimate its mean. And we had presumed access to the samples of this random vector X itself. So, in this sense, this problem was easy because we wanted to estimate something and we had access to the quantity of interest.



The above example was easy:

- Wanted to estimate $\mathbb{E}X$
- Had access to samples of X

However, in several real-world applications, it would be the case that

- we want to estimate $\mathbb{E}X$
- have access to samples of Y, which relates to X

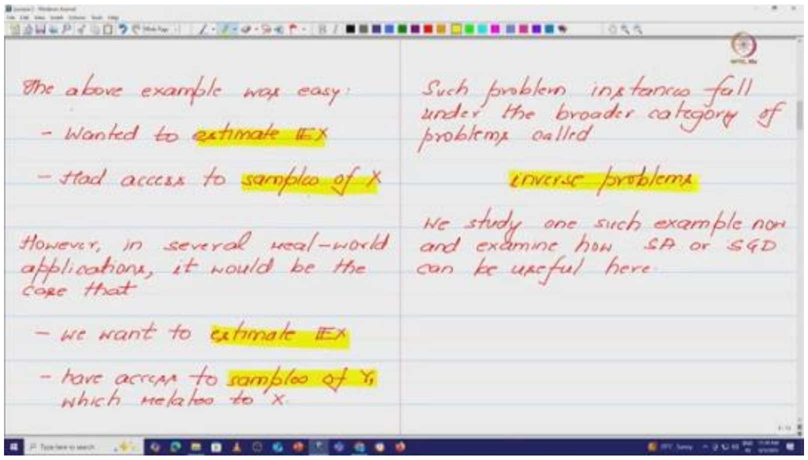
Such problem instances fall under the broader category of problems called

inverse problems

We study one such example now and examine how SA or SGD can be useful here

Direct access to the samples of the quantity of interest. However, in several real-world applications, it would be the case that we want to estimate the expected value of X—that is, the expected value of some random vector X. However, we do not have access to the samples of X. Instead, we only have access to the samples of a random vector Y, which in some sense relates to X. So, this is relatively more difficult than the problem we saw in the first example. The difficulty is that we want to estimate the expected value of X, but we do not have direct access to the samples of X.

Instead, we have access to samples of another vector Y, which in some sense relates to X. Such problem instances fall under the broad category of problems called inverse problems. We will now look at one such example and examine how stochastic approximation or stochastic gradient descent can be useful in such scenarios as well. So, this is Example 2. Again, the purpose of this example is to illustrate the utility of stochastic approximation algorithms in applications of interest.



The above example was easy:

- Wanted to estimate $\mathbb{E}X$
- Had access to samples of X

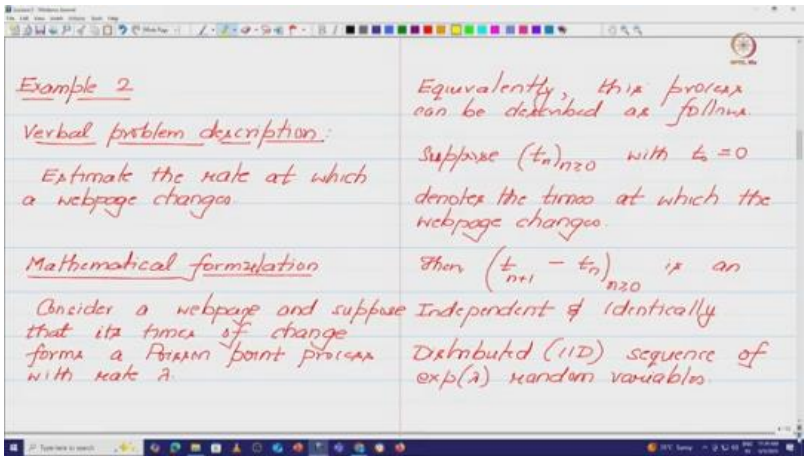
However, in several real-world applications, it would be the case that

- we want to estimate $\mathbb{E}X$
- have access to samples of Y, which relates to X

Such problem instances fall under the broader category of problems called

inverse problems

We study one such example now and examine how SA or SGD can be useful here



Example 2

Verbal problem description:

Estimate the rate at which a webpage changes.

Mathematical formulation

Consider a webpage and suppose that its times of change forms a Poisson point process with rate λ.

Equivalently, this process can be described as follows.

Suppose $(t_n)_{n\geq 0}$ with $t_0 = 0$ denotes the times at which the webpage changes.

Then $\left(t_{n+1} - t_n\right)_{n\geq 0}$ is an Independent & Identically Distributed (IID) sequence of $\exp(\lambda)$ random variables.

So first, let me give a simple verbal description of the problem that we plan to tackle under this example. The problem is that we have a web page. And we wish to estimate the rate at which this web page changes. So, let me elaborate a bit. Imagine your favorite web page.

Maybe if you are interested in cricket, there is a web page that displays scores and so on. Now, as new information about cricket becomes available, the web page updates itself. And we would like to know when this web page changes. In particular, we would like to know the rate at which this web page changes. So, what do I mean by that?

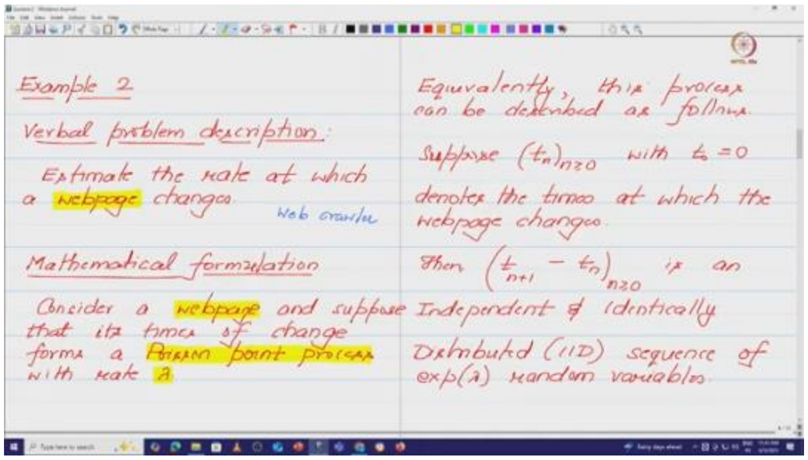
You can consider any unit interval of time that is of interest to you. For example, you could consider 1 second, 1 minute, or 1 hour—whatever works for you. And you would like to know how many times, on average, this web page changes in this desired unit interval of time. Now, one can ask: why would one be interested in knowing the rate at which a web page changes? Well, a simple motivating application would be in the context of what is called a web crawler.

So, sure in today's day and time, one uses maybe the LLM such as ChatGPT or Gemini and so on and so forth to ask questions. However, a few years back, we relied on search engines such as Google, Bing and so on and so forth to get answers to the questions that we may have had. So when I was young and I used a search engine, you know, I was very surprised. I was surprised because whenever I used to type in a question, the search engines used to instantaneously provide answers to these questions. And I always used to wonder, wow, how can this search engine actually search so quickly and fetch the answer for me?

So, over a period of time I realized that these search engines do not actually search at real time instead what they do is they keep crawling different web pages. So, by crawling I mean they keep visiting different web pages and checking whether the web page has changed or not and if it has changed download the latest information that is available to a local repository. And whenever a user actually provides a search query, the search engine doesn't search at that time. Instead, it searches within the local repository and in some sense instantaneously provides an answer to this question. So you can see that if you wanted to design a web crawler, you would need to know the rate at which different web pages change so that you can, you know, visit those pages, you know, at suitable intervals of time in order to fetch the latest information.

So, if you are interested in designing a web crawler, this question would be one of the fundamental questions of interest. So, let me repeat this question again. The goal in this question is to estimate the rate at which a web page changes. So, this is a verbal description, so now let us give a mathematical formulation and since So, this is not the main topic of this course. I will keep the mathematical formulation of this question in some simplified way, but you know one can build upon this idea and look at more sophisticated versions of this question as well.

So, what is the mathematical formulation of this question? We have a web page. We have a web page, and we will presume that the times at which this web page changes form what is called a Poisson point process with rate lambda. Is that okay? So, I will, you know, describe to you what a Poisson point process is.



Example 2

Verbal problem description:

Estimate the rate at which a webpage changes

Web crawler

Mathematical formulation

Consider a webpage and suppose that its times of change forms a Poisson point process with rate λ

Equivalently, this process can be described as follows.

Suppose $(t_n)_{n \geq 0}$ with $t_0 = 0$ denotes the times at which the webpage changes.

Then $(t_{n+1} - t_n)_{n \geq 0}$ is an Independent & Identically Distributed (IID) sequence of $\exp(\lambda)$ random variables

So, you can see that the times at which the web page changes is some stochastic process, and one of the simplest stochastic processes is what is called a Poisson point process, and this Poisson point process is completely characterized by the single scalar variable lambda. And, in some sense, this dictates the rate at which the web page changes, and the goal of our problem is to estimate lambda. So, before I discuss how we go about estimating this lambda, let me first try to explain what a Poisson point process means. A Poisson point process can be equivalently described as follows. Suppose Tn, n greater than or equal to 0.

So, this is a sequence of time instances—that is, you have T0, T1, T2, T3, and so on—and for simplicity, we will presume that T0 is 0. So, suppose that this sequence of time instances denotes the times at which the web page changes. So, we will say this sequence of time instances follows a Poisson point process if the interarrival time instances, which is Tn plus 1 minus Tn, So, if Tn is a real number and Tn plus 1 is a real number, then Tn plus 1 minus Tn will be another real number, and, of course, it goes without saying that these will be positive real numbers. So, you look at T1 minus T0, then T2 minus T1, T3 minus T2, and so on.

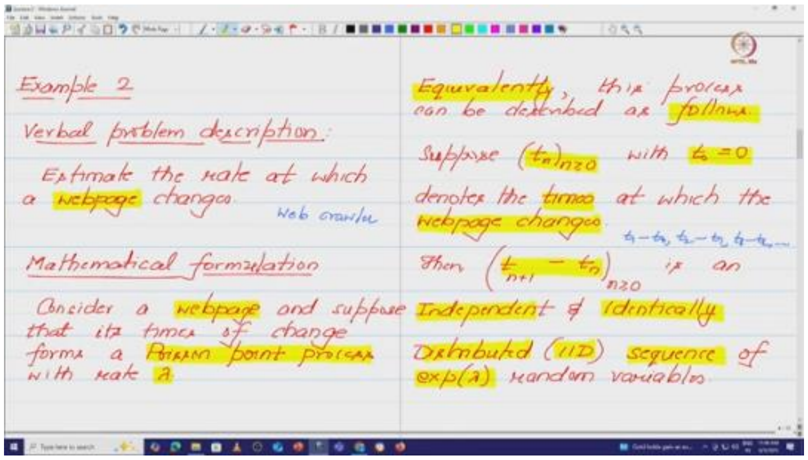
$$(t_n)_{n\geq 0}, \;\; t_0 = 0$$

$$(t_{n+1} - t_n)_{n\geq 0}$$

$$t_1 - t_0, t_2 - t_1, t_3 - t_2, \ldots$$

So, you look at the sequence. We will say that this sequence Tn is a Poisson point process if the sequence of interchange time instances are independent and identically distributed exponential random variables with parameter lambda. So, I will denote exponential random variables using the notation exp, and lambda over here denotes the rate of this exponential random variable. In other words, 1 over lambda is the expected value of this random variable. And you can already see that I have used the abbreviation IID to indicate independent and identically distributed.

So, this is a very common phrase. I am sure most of you must already be aware of this, and I will be using this abbreviation quite often in this course. So, please be aware of that. So, let me summarize this slide. So, we will say the times at which the web page changes form a Poisson point process if the interchange times form a sequence of independent and identically distributed exponential random variables with parameter lambda.



Example 2

Verbal problem description:

Estimate the rate at which a webpage changes.

Web crawler

Mathematical formulation

Consider a webpage and suppose that its times of change forms a Poisson point process with rate λ

Equivalently, this process can be described as follows.

Suppose $(t_n)_{n\geq 0}$ with $t_0 = 0$ denotes the times at which the webpage changes.

$t_1 - t_0, t_2 - t_1, t_3 - t_2, \ldots$

Then $(t_{n+1} - t_n)_{n\geq 0}$ is an Independent & Identically Distributed (IID) sequence of $\exp(\lambda)$ random variables.

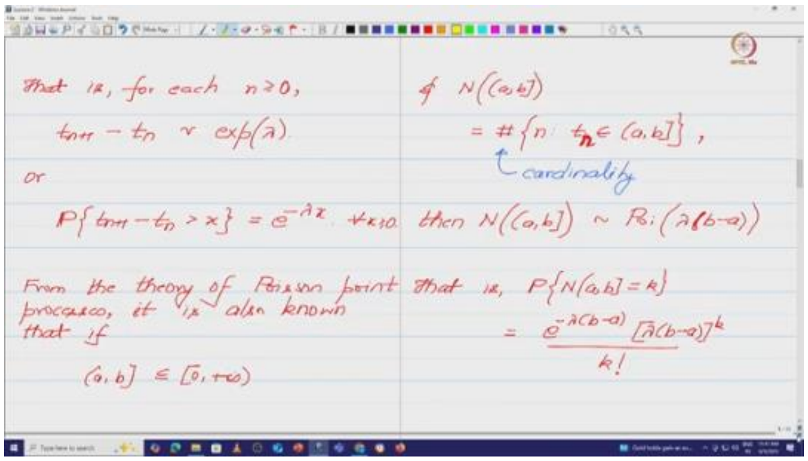
So, what does the interchange times being exponential mean? It means that for each n greater than 0, tn plus 1 minus tn. So, tn plus 1 minus tn is actually a random variable, and its distribution is like that of an exponential random variable with parameter lambda. Which means that if you look at the tail CDF of tn plus 1 minus tn, tail CDF means if you look at the probability that tn plus 1 minus tn is greater than or equal to x, this would equal e raised to minus lambda x for all x greater than 0.

$$n \geq 0$$

$$t_{n+1} - t_n \sim \exp(\lambda)$$

$$\mathbb{P} = \{t_{n+1} - t_n > x\} = e^{-\lambda x}\ \forall x \geq 0$$

So, you can see that this lambda appears over here, and this lambda is the same lambda that you have over here.



That is, for each $n \geq 0$,

$t_{n+1} - t_n \sim \exp(\lambda)$

or

$P\{t_{n+1} - t_n > x\} = e^{-\lambda x} \quad \forall x \geq 0$

From the theory of Poisson point processes, it is also known that if

$(a, b] \subseteq [0, +\infty)$

& $N((a,b])$

$= \#\{n : t_n \in (a, b]\}$,

cardinality

then $N((a,b]) \sim \text{Poi}(\lambda(b-a))$

that is, $P\{N(a,b] = k\}$

$= \dfrac{e^{-\lambda(b-a)} [\lambda(b-a)]^k}{k!}$

Right. And, you know, if you are aware of a bit of Poisson point processes, then we can also say much more about this—you know, times at which the web page changes. For example, we can say that, if you look at the interval A, B of the non-negative real line, right? Then the number of time instances that fall within [A, B], right? So, which we denote by N of the interval [A, B].

$$(a, b] \leq [o, +\infty)$$

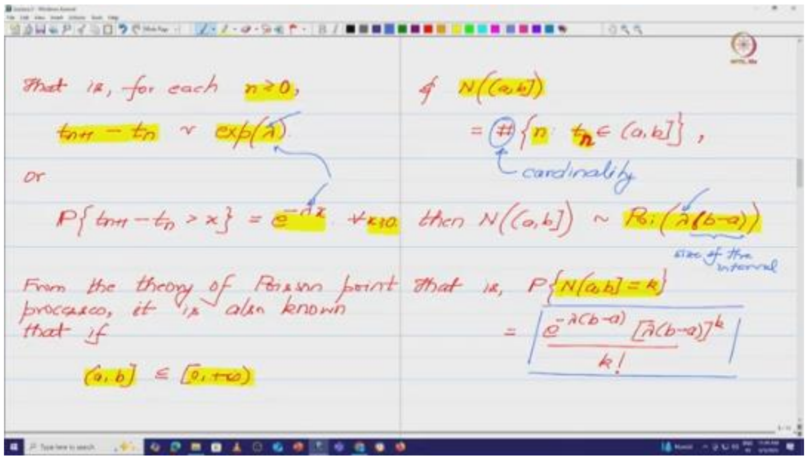
$$N\big((a, b]\big) = \#\{n: t_n \in (a, b]\},$$

This is basically the number, okay? So, the hash over here. denotes cardinality, And this set includes all those time instances N such that T_N is between A and B, or T_N lies in the interval [A, B]. And this interval I denote it as you know open from A side and closed at B side but it does not matter. Because these exponential random variables are continuous, so it does not matter how I define these sets. But as per the standard notation, I will work with such sets that are open from the left and closed from the right.

So, if you let n of the interval a, b be the number of time instances that lie in a, b then one can check that n of a, b is actually a Poisson random variable with parameter lambda times b minus a. So, this lambda over here comes from this parameter lambda, and this (B minus A) is basically the size of the interval. And when I say this is a Poisson random variable, what I mean is that the probability that N of [A, B] equals K will equal this expression that is given over here. Alright, so this, in some sense, is a summary of the Poisson point process, and the name Poisson comes because of the fact that N of [A, B] has this Poisson distribution, right? So, let me now summarize what is the goal of this exercise, right?

$$N\big((a,b]\big) \sim P_o : \big(\lambda(b-a)\big)$$

$$P\{N(a,b] = R\} = \frac{e^{-\lambda(b-a)}[\lambda(b-a)]^k}{k!}$$



That is, for each $n \geq 0$,

$t_{n+1} - t_n \sim \exp(\lambda)$

or

$P\{t_{n+1} - t_n > x\} = e^{-\lambda x} \quad \forall x \geq 0.$

From the theory of Poisson point processes, it is also known that if

$(a, b] \subseteq [0, +\infty)$

& $N((a,b])$

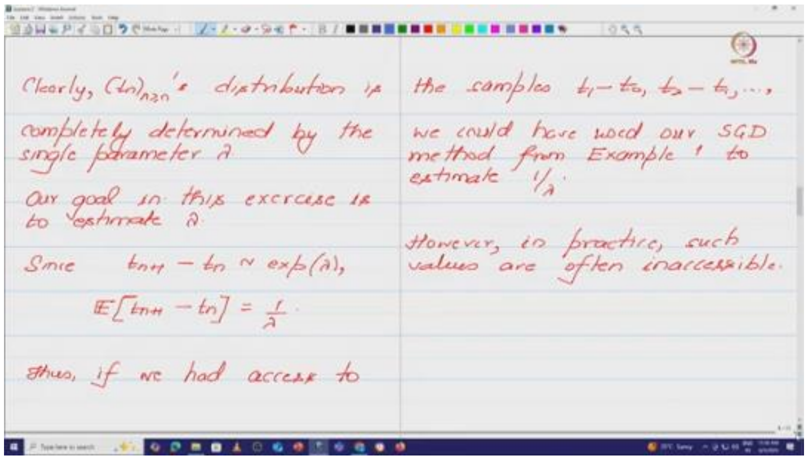
$= \#\{n : t_n \in (a,b]\},$

cardinality

then $N((a,b]) \sim Po : (\lambda \underbrace{(b-a)}_{\text{size of the interval}})$

that is, $P\{N(a,b] = k\}$

$= \boxed{\frac{e^{-\lambda(b-a)}[\lambda(b-a)]^k}{k!}}$



Clearly, $(t_n)_{n \geq 0}$'s distribution is completely determined by the single parameter λ

Our goal in this exercise is to estimate λ.

Since $t_{n+1} - t_n \sim \exp(\lambda)$,

$\mathbb{E}[t_{n+1} - t_n] = \frac{1}{\lambda}.$

Thus, if we had access to the samples $t_1 - t_0, t_2 - t_1, \ldots,$

we could have used our SGD method from Example 1 to estimate $1/\lambda$.

However, in practice, such values are often inaccessible.

Clearly, Tn's distribution is completely characterized by the single parameter lambda. So you can see that, to define this Poisson point process, I only need to specify this lambda, and once I specify this lambda, the inter-arrival times have a specific distribution, which is exponential with rate lambda. So that gets fixed. Right, and our goal in this exercise is to estimate lambda. Okay, that is the goal in this example, right?

So now, let's spend some time understanding how this example differs from the previous example. So recall in the previous example, we had a random vector X, and our goal was to estimate the expected value of X. And we presumed access to samples of the random vector X itself. So here, our goal is to estimate lambda. And let us first see how lambda is related to Tn plus 1 minus Tn.

So, as I said, lambda is the rate of these exponential random variables. Recall that Tn plus 1 minus Tn is exponential with rate lambda, which implies if you look at the expected value of Tn plus 1 minus Tn, then this will be 1 over lambda. So, in some sense, we want to estimate lambda, which can be equivalently posed as wanting to estimate 1 over lambda. So, this then has a relation with the problem that we looked at in Example 1, that is, if we had access to the values of T1 minus T0, T2 minus T1, and so on, right? So recall, a few slides back, I said that these inter-arrival times are exponential with rate lambda.

In other words, these are exponential with mean 1 over lambda. So, if we want to estimate 1 over lambda and we had access to T1 minus T0, T2 minus T1, and so on, then we could have used our SGD method from Example 1 to estimate 1 over lambda. And once we have 1 over lambda, you can invert that and get an estimate of lambda itself. So if we had access to T1 minus T0, T2 minus T1, and so on and so forth, we could have used the idea from the previous example. So recall that idea.

It was computing Xn plus 1 equals Xn plus alpha n. times capital Xn plus 1 minus Xn, right? So, this is the sample of interest and in this case what I will do is I will basically replace this by Tn plus 1 minus Tn whatever is this value I will replace it over here, and one can then show that this algorithm will actually converge to the mean of this quantity over here and the mean of this quantity we know is 1 over lambda. Hence, one can show that this little xn will actually converge to 1 over lambda. So, if you run this algorithm for long enough, one can see that xn will give you an approximate estimate of 1 over lambda, and if you invert it, you will get an estimate of lambda, which was the quantity of interest for us.

$$t_{n+1} - t_n \sim \exp(\lambda)$$

$$\mathbb{E}[t_{n+1} - t_n] = \frac{1}{\lambda}$$

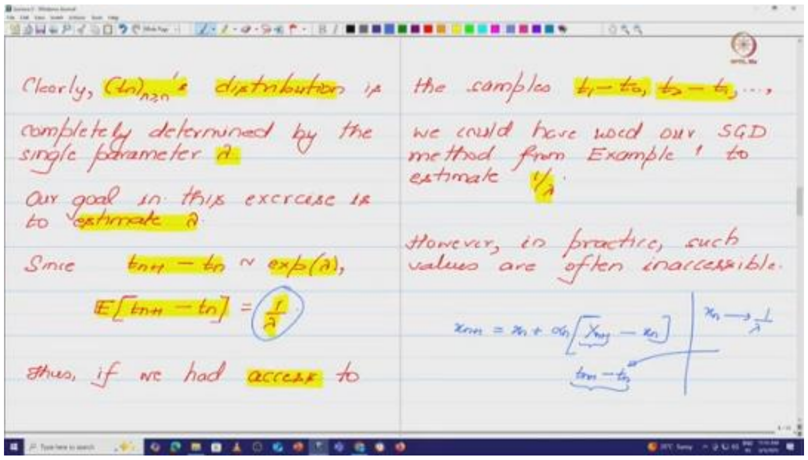
$$t_1 - t_0, t_2 - t_1, \dots$$

$$x_{n+1} = x_n + \alpha_n [X_{n+1} - x_n]$$

$$t_{n+1} - t_n$$

$$x_n \to \frac{1}{\lambda}$$

However, in practice, such values—by such values, I mean t1 minus t0, t2 minus t1, and so on and so forth—they will often be inaccessible. And the question is, in such a scenario, how can we estimate lambda? So, the question that one can ask is, if you cannot access t1 minus t0, t2 minus t1, and so on, what can we instead access, right? So, you know, this is like a practical difficulty - no web page may share, you know, the times at which it changes and so on and so forth, right? Instead, We can get to know if a web page has changed or not between two successive web page accesses.



Clearly, $(t_n)_{n \geq n}$'s distribution is completely determined by the single parameter λ.

Our goal in this exercise is to estimate λ.

Since $t_{n+1} - t_n \sim \exp(\lambda)$,

$$\mathbb{E}[t_{n+1} - t_n] = \boxed{\frac{1}{\lambda}}$$

Thus, if we had access to the samples $t_1 - t_0$, $t_2 - t_1$, …, we could have used our SGD method from Example 1 to estimate $1/\lambda$.

However, in practice, such values are often inaccessible.

$$x_{n+1} = x_n + \alpha_n [\underbrace{X_{n+1}}_{t_{n+1} - t_n} - x_n]$$

$$x_n \to \frac{1}{\lambda}$$

So, this is a practical suggestion. So, imagine you had a web crawler. It visited the web page at certain time instance. So it can download the current copy of that web page and then it can visit that web page at some other future time instance. And at that time, again, it can, you know, download the, you know, the later copy of that webpage.

So it has two copies of the webpage, one that it had access previously and one that it accessed in the future time instance. Right. Now this crawler or this you know software that you design can compare these two web pages and decide if the two copies differ and based on that conclude that yes you know between these two successive time successive web page accesses the web page indeed has changed right. So formally, if we denote Sn n greater than equal to 0 as the sequence of time instances at which we access the web page, then at time instance Sn plus 1, we will get to know if the web page is different from the copy that we had at time Sn. So, this comparison we can do and based on that we get to know whether the webpage has changed or not.

$$(s_n)_{n\geq 0}$$

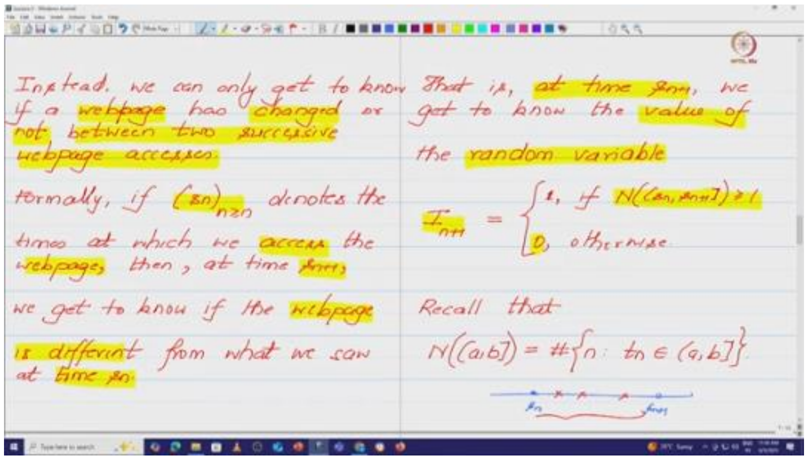
$$s_{n+1}$$

More formally, at time sn plus 1, we get to know the value of the random variable in plus 1. This random variable is 1 or 0. It is 1, if between SN and SN plus 1 there was at least one time instance where the web page changed. So more formally IN plus 1 will equal 1 if you know N of SN comma SN plus 1 that is the number of changes in this interval SN to SN plus 1 was greater than equal to 1 and 0 otherwise. So, what I am saying is that if the future copy and the previous copy do not differ from each other then you can conclude that the web page has not changed.

On the other hand if the future copy and the previous copy differ then one can conclude that yes the web page has changed. So, you can see that this value of IN plus 1 is something that we can get hold of. And the question that we would like to ask is, given access to the values of IN plus 1, right? When I say values of IN plus 1, I mean given access to I1, I2, I3, I4, and so on and so forth. Can we get hold of an estimate of lambda, right?

Now, the challenge with IN plus 1 in some sense is that between two successive visits, right, if you sort of imagine this to be the timeline, and let us say you visited a web page at this time and you visited this web page again at s_{n+1}. Now in-between these time instances the web page could have changed once, it could have changed twice, it could have changed three times, and so on and so forth. But you know this IN plus one will not tell you how many times it has changed instead it will only tell you how if it has changed or not. So, you can see that IN plus 1 actually, you know, only gives you partial information about the web page change process, and using this partial information, our goal is to estimate the value of lambda. So, I will conclude this lecture at this point by summarizing what we have discussed, and in the next class, we will, you know, discuss another stochastic approximation algorithm that makes use of the values of IN plus 1 to get hold of an estimate of lambda.

$$I_{n+1} = \begin{cases} 1, if\ N((s_n, s_{n+1}]) \geq 1 \\ 0, \qquad\qquad otherwise \end{cases}$$



Instead, we can only get to know if a webpage has changed or not between two successive webpage accesses.

Formally, if $(s_n)_{n\geq n}$ denotes the times at which we access the webpage, then, at time s_{n+1}, we get to know if the webpage is different from what we saw at time s_n.

That is, at time s_{n+1}, we get to know the value of the random variable

$$I_{n+1} = \begin{cases} 1, & \text{if } N((s_n, s_{n+1}]) \geq 1 \\ 0, & \text{otherwise} \end{cases}$$

Recall that

$$N((a,b]) = \#\{n : t_n \in (a,b]\}$$

So, let me summarize what we have discussed in this class. In the first class - I mean in the - sorry not the first class - in the beginning of this class, we you know showed that the example one that we considered, which involved estimating the mean of a random vector x using samples of x itself, that update rule that we came up with was basically a stochastic gradient descent. And we said that that problem instance was relatively easy because, you know, that problem instance, you know, involved estimating expected value of X using samples of X itself. Then we said okay often in real life you know we may not be in such a good scenario instead we may want to estimate expected value of X using samples of another variable Y that depends on X. And we saw an example of one such scenario wherein we wanted to estimate the rate at which the web page changes.

But we only have access to these 01 random variables IN plus 1, which does not tell you how many times the web page changes between two successive visits, but rather it only tells you whether the web page has changed or not between these successive visits.

So, in the next class, we will see how we can make use of a stochastic approximation algorithm, use values of i n plus 1, and come up with a way to estimate lambda. Thank you. Namaste.