STOCHASTIC APPROXIMATION: THEORY AND APPLICATIONS

Dr. Gugan Thope

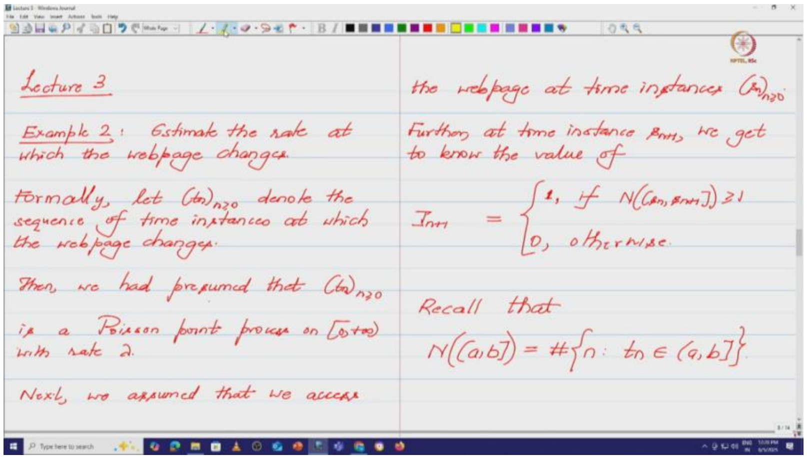
Department of Computer Science and Automation

Indian Institute of Science

Lecture 3

Estimating change rates of webpages

Hello and Namaste, everyone. Welcome to Lecture 3. Let us do a quick recap of what we have covered in Lectures 1 and 2. In Lecture 1, we discussed an example of stochastic approximation, which involved estimating the mean of a random vector X using samples of the same random vector X. Then, in Lecture 2, we began looking at a more realistic setup where the goal was to estimate the expected value of some random vector X using samples of another random vector Y.



Lecture 3

Example 2 : Estimate the rate at which the webpage changes.

Formally, let $(t_n)_{n \geq 0}$ denote the sequence of time instances at which the webpage changes.

Then, we had presumed that $(t_n)_{n \geq 0}$ is a Poisson point process on $[0, +\infty)$ with rate λ.

Next, we assumed that we access the webpage at time instances $(s_n)_{n \geq 0}$.

Further, at time instance s_{n+1}, we get to know the value of

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

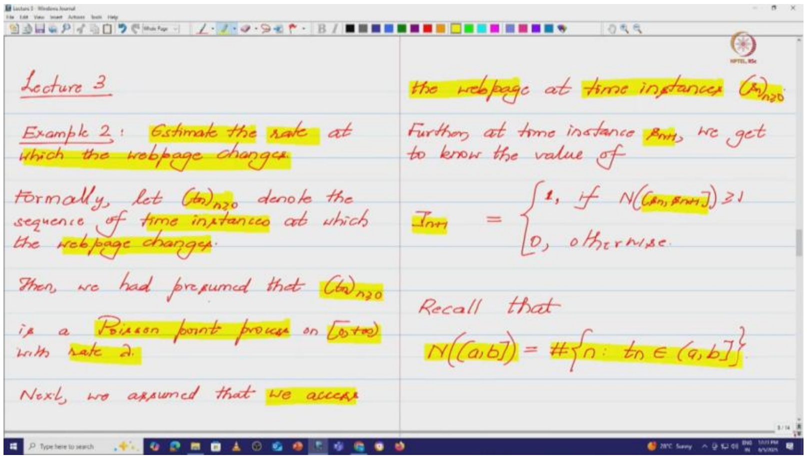
Recall that

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

More specifically, we looked at the problem of estimating the rate at which the web page changes. So, this was the problem that we considered. And formally, we described this problem in the following way. We presume that Tn, the sequence Tn—that means T0, T1, T2, T3, and so on and so forth—denotes the time instances at which the web page changes. And then, for simplicity, we had presumed that this sequence forms a Poisson point process on the positive or non-negative real line with rate lambda.

And I had told you that the sequence Tn is said to be a Poisson point process if the interarrival times—that is, Tn+1 minus Tn—for different values of n, form a sequence of independent and identically distributed exponential random variables with parameter lambda. Right. And then we said that we don't have direct access to the times at which the web page changes. Instead, we said that often we may only be able to access the web page at time instances SN. Right, and at time instance SN+1, we only get to know if the web page has changed in the interval SN to SN+1.

in particular, at time instance SN plus 1, we said that we have access to the value of the random variable IN plus 1 which is a 01 random variable, 0, if the web page has not changed between SN and SN plus 1 and 1, if it has changed at least once. So, recall that NAB equals the number of values in the set described over here. So, this set includes all those values of n where tn lies between a and b. So, n of ab is the cardinality of this set, and I denote cardinality by this hash variable, right?



Lecture 3

Example 2: Estimate the rate at which the webpage changes.

Formally, let $(t_n)_{n\geq 0}$ denote the sequence of time instances at which the webpage changes.

Then, we had presumed that $(t_n)_{n\geq 0}$ is a Poisson point process on $[0,+\infty)$ with rate λ.

Next, we assumed that we access the webpage at time instances $(s_n)_{n\geq 0}$

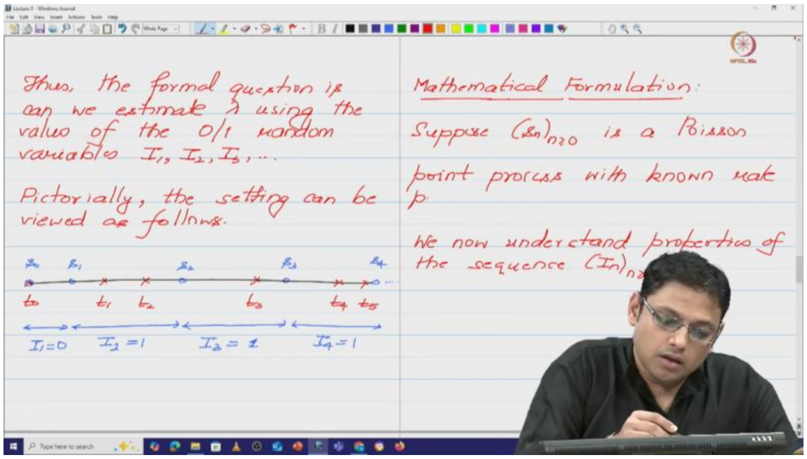
Further, at time instance s_{n+1}, we get to know the value of

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

Recall that

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

So, this is what I use over here, this definition is what I use over here, right? And notice that Sn, Sn plus 1, these are all random variables, right? And n of Sn comma Sn plus 1 is another random variable, right? So, now our formal question is, can we estimate lambda which denotes the rate at which the web page changes using values of the 01 random variables i1, i2, i3 and so on and so forth. So, let us get a pictorial view of this problem.



Thus, the formal question is can we estimate λ using the values of the 0/1 random variables $I_1, I_2, I_3, \ldots$

Pictorially, the setting can be viewed as follows.

Mathematical Formulation:

Suppose $(s_n)_{n\geq 0}$ is a Poisson point process with known rate ρ.

We now understand properties of the sequence $(I_n)_{n\geq}$

So, imagine this black line to denote the time axis and the red crosses. So, you can see that there are a bunch of red crosses. So, let us presume that these red crosses denote the times at which the web page changes. So, you can see that you know, so t0 is basically some initial point of convenience. So, do not read too much into T0, we will presume that T0 is 0, right.

So, T1 is the first time at which the web page changes from this chosen time of reference. T2 is the second time instance at which the web page changes. We can have similar interpretations for T3, T4, and T5. Notice that the distance between T0 to T1, T1 to T2, T2 to T3 is not the same. This is on purpose because, recall that if T0, T1, T2, T3, and so on form a Poisson point process.

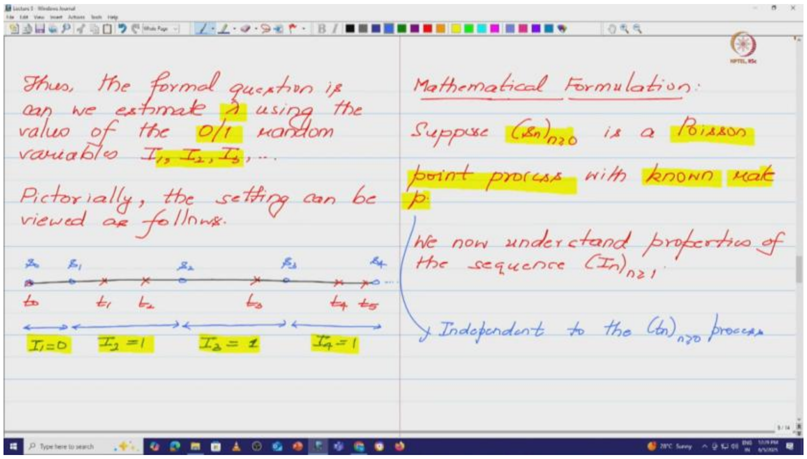
Then the interarrival times—that is, T1 minus T0, T2 minus T1, and so on—are themselves exponential random variables, right? And because they are random variables, the width of these intervals will be different, and each time you repeat this experiment, the positions of T0, T1, T2, T3, and so on will differ. For this particular instance of the experiment, let us presume that the places where I have put crosses are the times at which the web page changes. On the other hand, we have the time instances denoted by these blue circles at which we access the web page. So again, for convenience, we say that S0 is 0, and that matches T0.

So, from this reference time, we access this web page at times S1, S2, S3, and so on. And again, we can presume that S1 minus S0, S2 minus S1, and so on are regularly spaced. Recall that S0, S1, S2, and so on are under our control because we are designing a web crawler that visits this web page. So, we are designing this crawler, and we can choose how often it visits this web page—whether at periodic intervals or at random instances of time—all of which is under our control. On the other hand, T0, T1, T2, T3, and so on are not under our control; those are the times at which the web page changes, which is beyond our control.

But S0, S1, S2, S3, and so on, they are indeed under our control, right? Now, as I told you, I1 is the 0-1 random variable which tells us whether the web page has changed between S0 and S1. Now, in this case, you can see that there is no cross which is strictly bigger than S0 and less than or equal to S1, and that is the reason I1 is 0. On the other hand, I2 looks at whether the web page has changed between S1 and S2, and you can see that there are two crosses between S1 and S2, which means the web page changed twice. But I1 will only be 1 because we will look at the copy of the webpage at time instance S2 and compare it with the copy of the webpage that we had accessed at time instance S1, and we will only see that the webpage has changed.

Because of this reason, this I2 will only be 1 and not 2. And similarly, I3 is 1 because there is exactly one time instance between S2 and S3, which is T3, at which the web page changed. On the other hand, between S3 and S4, there are two time instances at which the web page changed; hence, I4 is again 1. So, the question is, given the values of I_1, I_2, I_3 ... which, again, notice that arrive sequentially—can we come up with a stochastic approximation algorithm to estimate lambda, right? Now, as I told you, the time instances at which we visit this web page are under our control, so we could choose them to be regularly placed.

However, for ease of analysis, we will presume that SN—that is, the sequence SN, which is the time instances at which we access the web page—is actually another Poisson point process. But we will presume that this is a Poisson point process with a known rate P. So, we will presume that, and I should add that this process—sorry. I should add that this process—we will presume—is independent of the TN process. So, now, under this assumption, our goal is to estimate lambda.



Thus, the formal question is can we estimate λ using the value of the 0/1 random variables $I_1, I_2, I_3, \ldots$

Pictorially, the setting can be viewed as follows.

s_0 s_1 s_2 s_3 s_4
t_0 t_1 t_2 t_3 t_4 t_5
$I_1=0$
$I_2=1$
$I_3=1$
$I_4=1$

Mathematical Formulation:

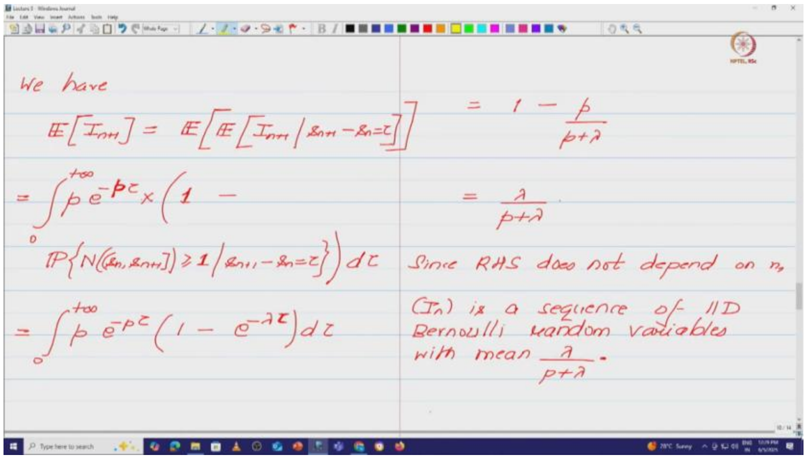
Suppose $(s_n)_{n\geq 0}$ is a Poisson point process with known rate p.

We now understand properties of the sequence $(I_n)_{n\geq 1}$.

Independent to the $(t_n)_{n\geq 0}$ process

So, before we discuss the algorithm to estimate lambda, what we will do is first estimate or first understand the properties of the sequence IN. So, we will do that now. So, recall that IN is actually a 0-1 Bernoulli random variable. So, let us try to compute its expected value. Now, by the properties of iterated expectation, one knows that the expected value of IN plus 1 is the expectation of the expectation of IN plus 1 under the condition that sn plus 1 minus sn equals tau.

$$\mathbb{E}[I_{n+1}] = \mathbb{E}\big[\mathbb{E}[I_{n+1}|s_{n+1} - s_n = \tau]\big]$$



We have

$$\mathbb{E}[I_{n+1}] = \mathbb{E}\Big[\mathbb{E}\big[I_{n+1} / s_{n+1} - s_n = \tau\big]\Big]$$

$$= \int_0^{+\infty} p\, e^{-p\tau} \times \Big(1 - \mathbb{P}\{N((s_n, s_{n+1}]) \geq 1 / s_{n+1} - s_n = \tau\}\Big)\, d\tau$$

$$= \int_0^{+\infty} p\, e^{-p\tau}\left(1 - e^{-\lambda\tau}\right) d\tau$$

$$= 1 - \frac{p}{p+\lambda}$$

$$= \frac{\lambda}{p+\lambda}.$$

Since RHS does not depend on n, (I_n) is a sequence of IID Bernoulli random variables with mean $\frac{\lambda}{p+\lambda}$.

So, recall that sn plus 1 minus sn, because of our choice that the sequence sn is a Poisson point process with rate lambda, sn plus 1 minus sn will be exponential with rate p, right? The times at which the web page changes is a Poisson point process with an unknown rate lambda. However, the times at which we access the web page is a Poisson point process with rate P. Hence, SN plus 1 minus SN is an exponential random variable with parameter

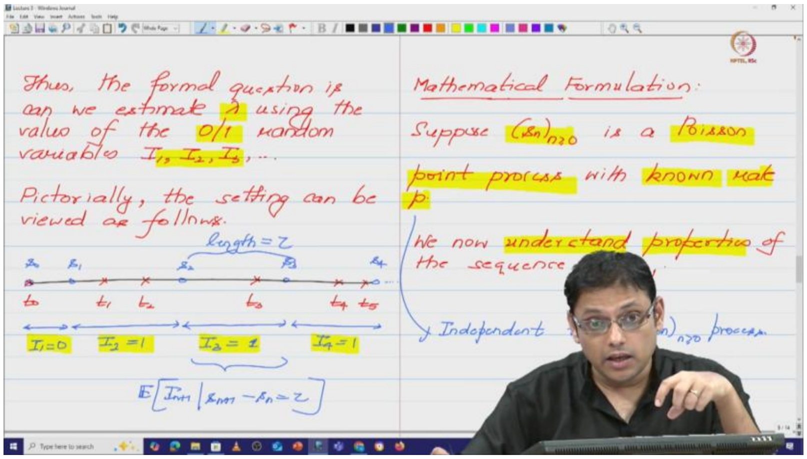
P. So, our goal is, given that SN plus 1 minus SN equals T, what is the conditional expectation of IN plus 1?

Now, recall that IN plus 1. So, let me go to the previous slide to explain this. So, recall that IN plus 1 tells us how many times the web page changes between two successive visits, in particular between SN and SN plus 1. And now, under this conditional expectation, what we are saying is that suppose this time interval is of length tau, okay? So, we are presuming that the length of time over here equals tau.

Given this information, you know, what is the expected value of I_n plus 1, which is equivalent to asking when I_n plus 1 equals 1, because I_n plus 1 is a Bernoulli random variable, right? So, we want to compute the expected value of I_N plus 1, and from the basic properties of expectation, we know that the expected value of I_N plus 1 can be written using the iterated expectation property, that is, the expectation of the expectation of I_N plus 1 given S_N plus 1 minus S_N equals tau.

$$\mathbb{E}[I_{n+1} | s_{n+1} - s_n = \tau]$$

Now, recall that because the times at which we access the web page form a Poisson point process with a known rate P, this S_N plus 1—sorry, this S_N plus 1 minus S_N—would actually be an exponential random variable with rate P, right? And now, let me tell you how to find this expectation. So, I have given you the expression over here.



Thus, the formal question is can we estimate λ using the value of the 0/1 random variables $I_1, I_2, I_3, \ldots$

Pictorially, the setting can be viewed as follows.

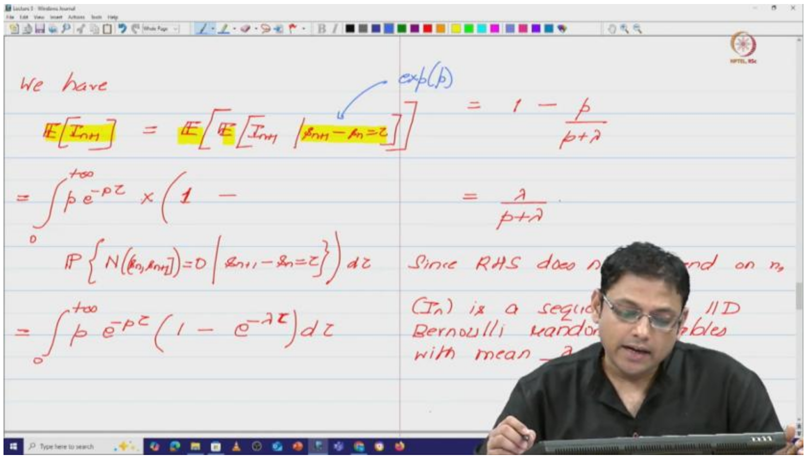
$\mathbb{E}[I_{n+1} | s_{n+1} - s_n = \tau]$

Mathematical Formulation:

Suppose $(s_n)_{n\geq 0}$ is a Poisson point process with known rate p.

We now understand properties of the sequence

Independent n)$_{n\geq 0}$ process.



We have

$$\mathbb{E}[I_{n+1}] = \mathbb{E}\left[\mathbb{E}\left[I_{n+1} \middle| \overset{\exp(p)}{s_{n+1} - s_n = \tau}\right]\right]$$

$$= \int_0^{+\infty} p e^{-p\tau} \times \left(1 - \mathbb{P}\{N((s_n, s_{n+1}]) = 0 \mid s_{n+1} - s_n = \tau\}\right) d\tau$$

$$= \int_0^{+\infty} p e^{-p\tau} (1 - e^{-\lambda\tau}) d\tau$$

$$= 1 - \frac{p}{p+\lambda}$$

$$= \frac{\lambda}{p+\lambda}.$$

Since RHS does n nd on n,

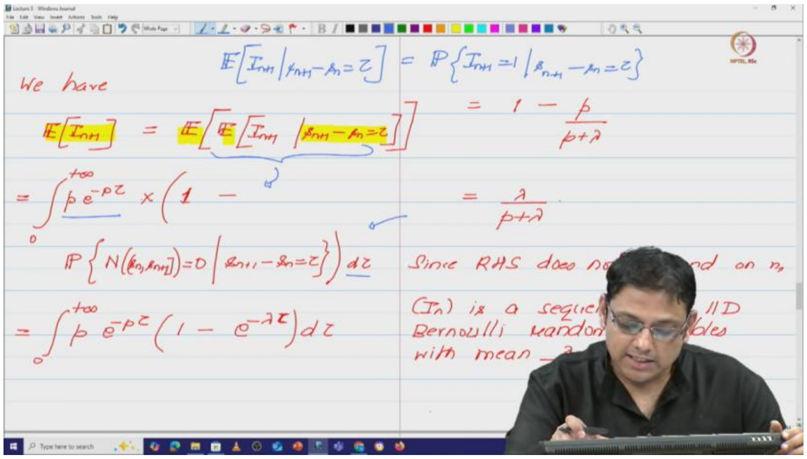
(I_n) is a seque IID Bernoulli random ables with mean λ

So, the outer expectation basically figures out what the probability is that S_n plus 1 minus S_n takes the value tau in a small interval. So, this is given over here—this times d tau—and this inner conditional expectation is given in this round bracket. So, let me now explain how this inner conditional expectation is found. So, the inner conditional expectation—because your I_N plus 1 is a Bernoulli random variable—actually equals the conditional probability that I_N plus 1 equals 1, given S_N plus 1 minus S_n equals tau.

Now, this probability equals 1 minus the probability that I_n plus 1 equals 0 with the same conditioning. The same conditioning that we have over here—place it over here. Now, I_n plus 1 is 0 if and only if there are no changes in the interval starting from S_n and ending at S_n plus 1. So, that is what I have written over here. So, this probability and this probability are one and the same.

$$\mathbb{E}[I_{n+1}|s_{n+1} - s_n = \tau] = \mathbb{P}\{I_{n+1} = 1|s_{n+1} - s_n = \tau\}$$

$$= 1 - \mathbb{P}\{I_{n+1} - 0|s_{n+1} - s_n = \tau\}$$



We have

$\mathbb{E}[I_{n+1} | s_{n+1} - s_n = \tau] = \mathbb{P}\{I_{n+1} = 1 | s_{n+1} - s_n = \tau\}$

$= 1 - \frac{p}{p+\lambda}$

$= \frac{\lambda}{p+\lambda}$

$\mathbb{E}[I_{n+1}] = \mathbb{E}[\underbrace{\mathbb{E}[I_{n+1} | s_{n+1} - s_n = \tau]}]$

$= \int_0^{+\infty} p e^{-p\tau} \times (1 - \mathbb{P}\{N((s_n, s_{n+1}]) = 0 | s_{n+1} - s_n = \tau\}) d\tau$

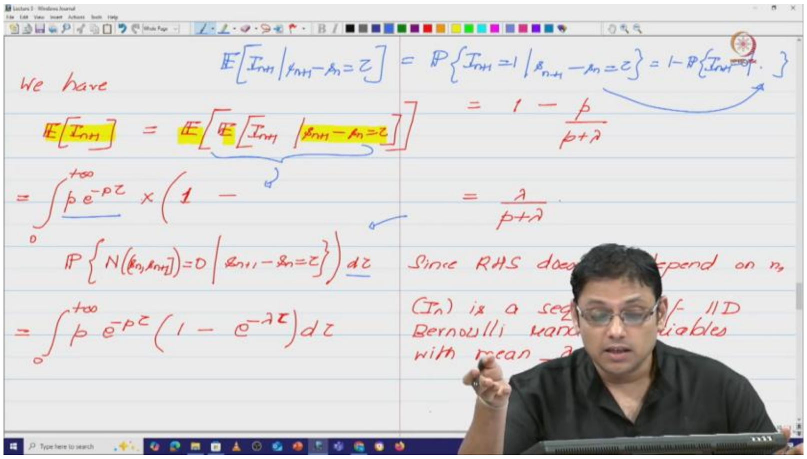
$= \int_0^{+\infty} p e^{-p\tau} (1 - e^{-\lambda\tau}) d\tau$

Since RHS does no nd on n,

(I_n) is a seque IID

Bernoulli random bles

with mean λ



We have

$\mathbb{E}[I_{n+1} | s_{n+1} - s_n = \tau] = \mathbb{P}\{I_{n+1} = 1 | s_{n+1} - s_n = \tau\} = 1 - \mathbb{P}\{I_{n+1} = 0 .\}$

$= 1 - \frac{p}{p+\lambda}$

$= \frac{\lambda}{p+\lambda}$

$\mathbb{E}[I_{n+1}] = \mathbb{E}[\underbrace{\mathbb{E}[I_{n+1} | s_{n+1} - s_n = \tau]}]$

$= \int_0^{+\infty} p e^{-p\tau} \times (1 - \mathbb{P}\{N((s_n, s_{n+1}]) = 0 | s_{n+1} - s_n = \tau\}) d\tau$

$= \int_0^{+\infty} p e^{-p\tau} (1 - e^{-\lambda\tau}) d\tau$

Since RHS doe pend on n,

(I_n) is a seq f IID

Bernoulli rand iables

with mean λ

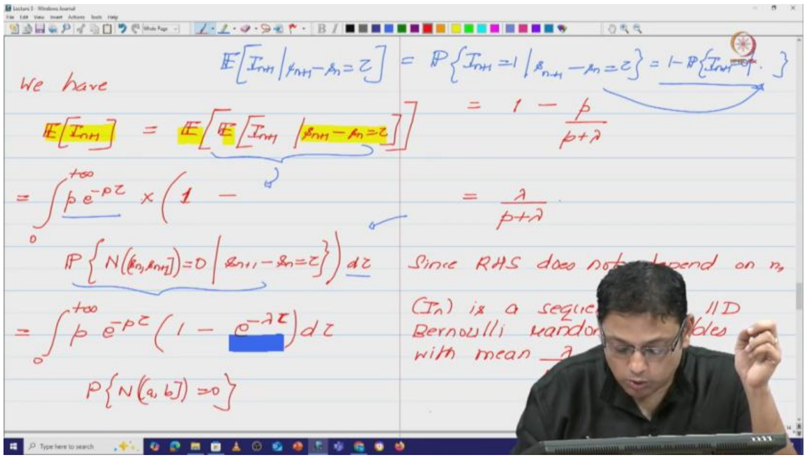
Right, and because, you know, the times at which your web page changes and the times at which, you know, we access the web page, these two processes are independent. One can show that this probability actually equals e raised to minus lambda tau, right? So this just follows from the fact that, you know, a Poisson random variable, right? Like, if you take this a b interval and if you ask what is the probability that this is 0. So because you have a Poisson random variable, this is e raised to minus lambda times b minus a, okay?

$$= \int_0^{+\infty} p\, e^{-p\tau} \times \left(1 - \mathbb{P}\{N((s_n, s_{n+1}]) = 0 | s_{n+1} - s_n = \tau\}\right) d\tau$$

$$= \int_0^{+\infty} p\, e^{-p\tau} \left(1 - e^{-\lambda\tau}\right) d\tau$$

$$\mathbb{P}\{N((a, b]) = 0\} = e^{-\lambda(b-a)}$$

So this is exactly what I have written over here, and this b minus a gets replaced by tau because of this conditioning over here.



We have

$$\mathbb{E}[I_{n+1}] = \mathbb{E}\Big[\underbrace{\mathbb{E}[I_{n+1} \mid s_{n+1} - s_n = \tau]}\Big]$$

$$= \int_0^{+\infty} \underline{p\, e^{-p\tau}} \times \Big(1 - \underbrace{\mathbb{P}\{N((s_n, s_{n+1}]) = 0 \mid s_{n+1} - s_n = \tau\}}\Big)\, \underline{d\tau}$$

$$= \int_0^{+\infty} p\, e^{-p\tau}(1 - e^{-\lambda\tau})\, d\tau$$

$$P\{N((a, b]) = 0\}$$

$$\mathbb{E}[I_{n+1} \mid s_{n+1} - s_n = \tau] = \mathbb{P}\{I_{n+1} = 1 \mid s_{n+1} - s_n = \tau\} = 1 - \mathbb{P}\{I_{n+1} = 0 \ldots\}$$

$$= 1 - \frac{p}{p+\lambda}$$

$$= \frac{\lambda}{p+\lambda}.$$

Since RHS does not depend on n,

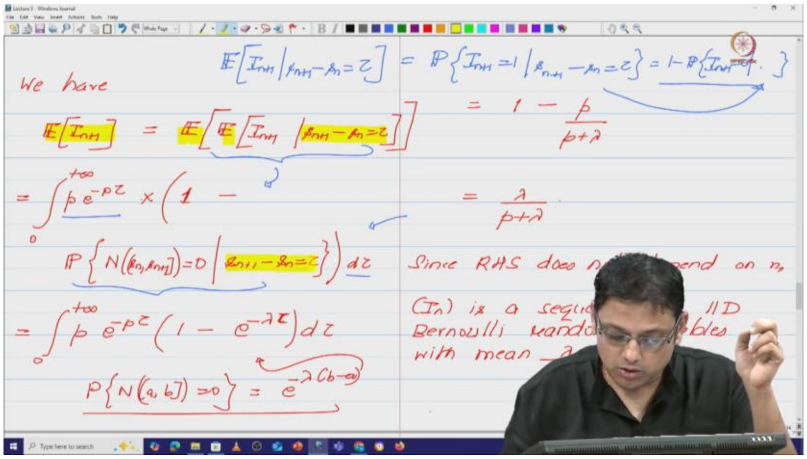
(I_n) is a seque[illegible] IID Bernoulli random [illegible]bles with mean $\frac{\lambda}{\ }$

Just a minute, let me try this again because of this conditioning over here, right. So, that is why you end up with this e raised to minus lambda tau, and this just follows from this expression. So, now let me erase all these quantities over here, all right. So, this integral, by a simple calculation, can be seen to be 1 minus p over p plus lambda, which if you rearrange and do a bit of algebra can be seen to be lambda over p plus lambda. So, recall that this is the expression for the expected value of I n plus 1.

$$= \int_0^{+\infty} p\, e^{-p\tau}\left(1 - e^{-\lambda\tau}\right) d\tau$$

$$= 1 - \frac{p}{p+\lambda}$$

$$= \frac{\lambda}{p+\lambda} = \mathbb{E}[I_{n+1}]$$



We have

$$\mathbb{E}[I_{n+1} | s_{n+1} - s_n = z] = \mathbb{P}\{I_{n+1} = 1 | s_{n+1} - s_n = z\} = 1 - \mathbb{P}\{I_{n+1} \cdot \}$$

$$= 1 - \frac{p}{p+\lambda}$$

$$= \frac{\lambda}{p+\lambda}$$

$$\mathbb{E}[I_{n+1}] = \mathbb{E}[\mathbb{E}[I_{n+1} | s_{n+1} - s_n = z]]$$

$$= \int_0^{+\infty} p e^{-pz} \times \left(1 - \mathbb{P}\{N((s_n, s_{n+1}]) = 0 | s_{n+1} - s_n = z\}\right) dz$$

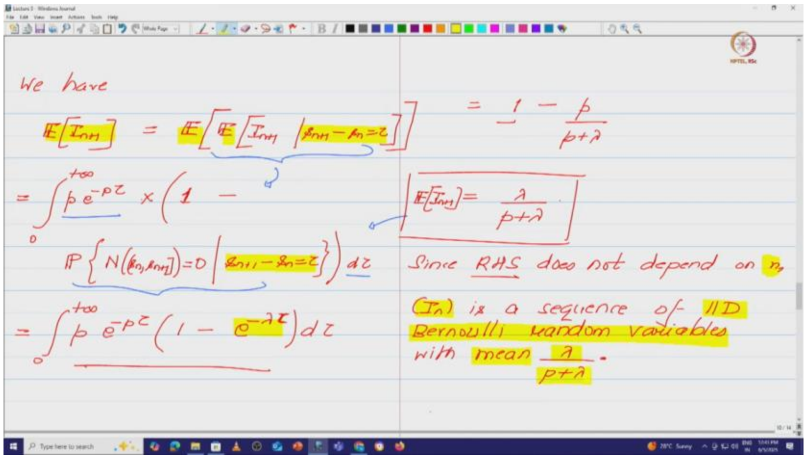
$$= \int_0^{+\infty} p e^{-pz} (1 - e^{-\lambda z}) dz$$

$$\mathbb{P}\{N((a, b]) = 0\} = e^{-\lambda(b-a)}$$

Since RHS does n… …pend on n,

(I_n) is a seque… IID Bernoulli rando… …bles with mean λ…

Now, since the right-hand side over here, which I abbreviate as RHS, does not depend on n, right? One can conclude that the sequence I N is actually a sequence of independent and identically distributed Bernoulli random variables with mean lambda over P plus lambda, which is the expression that we had found over here. So, now what we are going to do is we are going to make use of this observation and come up with an algorithm to find lambda. So, this is our goal now. So, let us see how we can go ahead and do that.



We have

$$\mathbb{E}[I_{n+1}] = \mathbb{E}[\mathbb{E}[I_{n+1} | s_{n+1} - s_n = z]]$$

$$= 1 - \frac{p}{p+\lambda}$$

$$\boxed{\mathbb{E}[I_{n+1}] = \frac{\lambda}{p+\lambda}}$$

$$= \int_0^{+\infty} p e^{-pz} \times \left(1 - \mathbb{P}\{N((s_n, s_{n+1}]) = 0 | s_{n+1} - s_n = z\}\right) dz$$

$$= \int_0^{+\infty} p e^{-pz} (1 - e^{-\lambda z}) dz$$

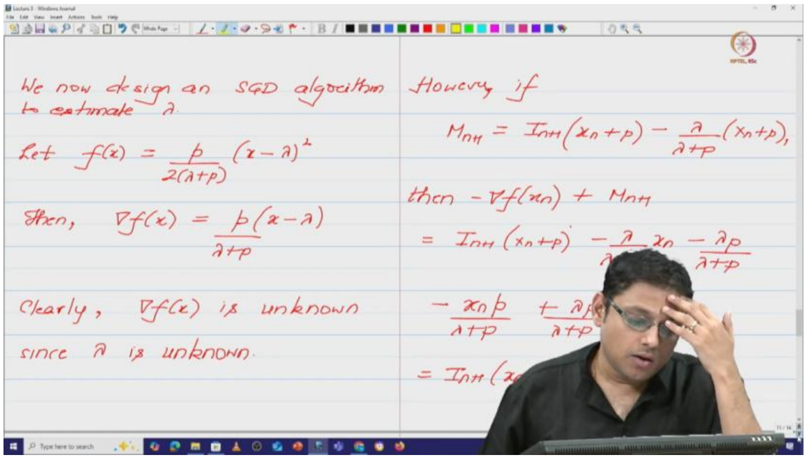
Since RHS does not depend on n,

(I_n) is a sequence of IID Bernoulli random variables with mean $\frac{\lambda}{p+\lambda}$.

So, now in order to come up with this algorithm, let me first define a function f. So, this function goes from the real line to the real line and is given by this definition over here. It is p over 2 times lambda plus p times x minus lambda the whole square.

$$f(x) = \frac{p}{2(\lambda + P)} (x - \lambda)^2$$

And so, how we came up with this function and so on I will talk about it, but at this point bear with me and let us say f is some R to R function given by this expression. So, if you look at grad f of x, then you can see that this expression is basically p over lambda plus p times x minus lambda.



We now design an SGD algorithm to estimate λ.

Let $f(x) = \frac{p}{2(\lambda+p)}(x-\lambda)^2$

Then, $\nabla f(x) = \frac{p(x-\lambda)}{\lambda+p}$

Clearly, $\nabla f(x)$ is unknown since λ is unknown.

However if

$M_{n+1} = I_{n+1}(x_n+p) - \frac{\lambda}{\lambda+p}(x_n+p),$

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

$= I_{n+1}(x_n+p) - \frac{\lambda}{\lambda} x_n - \frac{\lambda p}{\lambda+p}$

$- \frac{x_n p}{\lambda+p} + \frac{\lambda p}{\lambda+p}$

$= I_{n+1}(x$

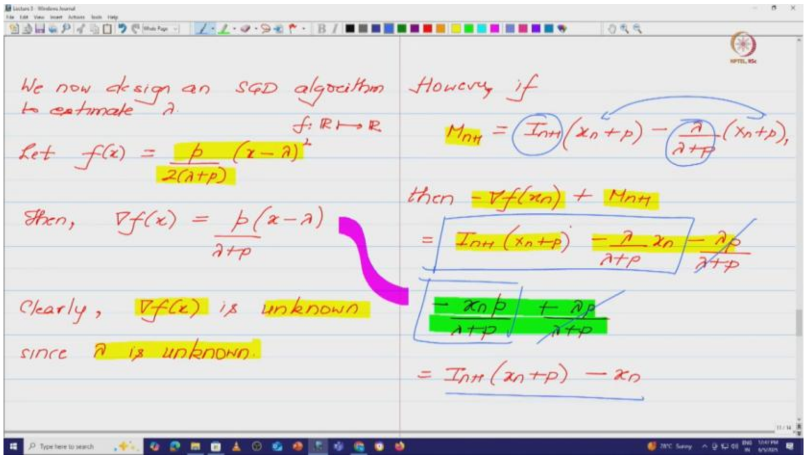
Now the expression for gradient of f of x is unknown right at any x since lambda is unknown. So we do not know lambda; hence, the value of grad f of x is unknown. If we define the noise sequence Mn plus 1 using the expression on the right, I will tell you how this noise expression came about and so on and so forth, but bear with me for the time being. If we define the noise expression using the following formula, that is In plus 1 times Xn plus P minus lambda over lambda plus P times Xn plus B. So, at this point you can just note that this In plus 1 that you have over here, its expected value is written over here, and these two expressions are basically one and the same.

So, you can sort of see the nice structure that Mn plus 1 has.

$$\nabla f(x) = \frac{p(x-\lambda)}{\lambda + P}$$

$$M_{n+1} = I_{n+1}(x_n + P) - \frac{\lambda}{\lambda + P}\,(X_n + P)$$

And now if you look at the expression for minus grad f of xn plus Mn plus 1, then you can see that Mn plus 1 from this above expression equals what is given over here. And the value of minus grad f of xn from this equals this expression over here. Right, and you can see that these two terms actually cancel off, and hence by simple algebra one can see that whatever remains—that is, this expression and this expression—if you combine them, you can see that we end up with this expression. So, at this point I am sure as a reader you would not know why we chose this f function, why we defined Mn plus 1 to be in this form.



We now design an SGD algorithm to estimate λ.

$f: \mathbb{R} \mapsto \mathbb{R}$

Let $f(x) = \frac{p}{2(\lambda+p)}(x-\lambda)^2$

Then, $\nabla f(x) = \frac{p(x-\lambda)}{\lambda+p}$

Clearly, $\nabla f(x)$ is unknown since λ is unknown.

However if

$$M_{n+1} = \boxed{I_{n+1}}(x_n+p) - \boxed{\frac{\lambda}{\lambda+p}}(x_n+p),$$

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

$$= \boxed{I_{n+1}(x_n+p) - \frac{\lambda}{\lambda+p}x_n} - \cancel{\frac{\lambda p}{\lambda+p}}$$

$$\boxed{-\frac{x_n p}{\lambda+p}} \cancel{+\frac{\lambda p}{\lambda+p}}$$

$$= I_{n+1}(x_n+p) - x_n$$

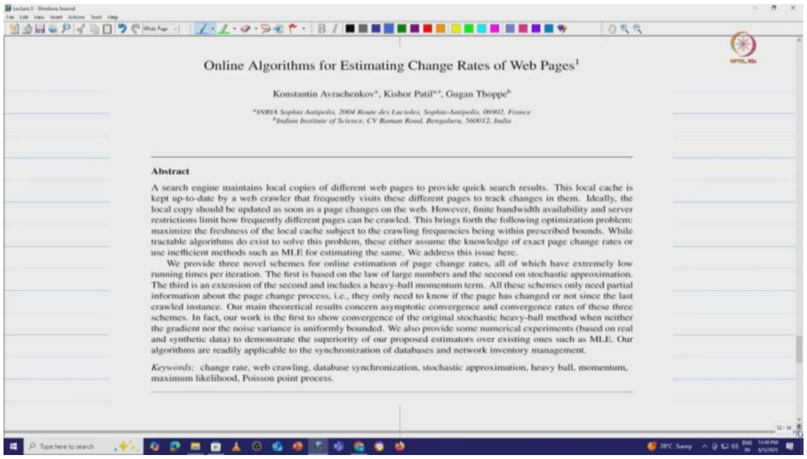
However, what you can observe is this expression for minus grad f of xn plus mn plus 1 is what is given over here. And this expression can be computed at the time step n. It can be computed because at time step n, which in the real line corresponds to the true time instance Sn plus 1, we would know the value of In plus 1. And since we would have already computed Xn, we would also know Xn and P is the rate at which we access the webpage. So, this is also something that we know.

So, one can see that every term in this blue box is actually known. Hence, this expression is known, right? So, while grad of f of Xn is unknown because lambda is unknown. The value of minus grad f of xn plus mn plus 1 is known. In other words, while the required gradient is unknown, a noisy estimate of this gradient can be obtained by just looking at the values of in plus 1, xn and p.

So, this is in some sense the advantage of working with this f and this noise variable. You know you can actually get hold of more details about this algorithm which I had the opportunity to work on as part of this paper with my collaborators Professor Konstantin Averachenkov and Kishore Patil. This is the name of the paper in which we had designed this algorithm and if you are interested you can go to my Google Scholar profile and you know get hold of this paper and read the details further. So, at this point what we have is that while minus grad f of xn is unknown because lambda is unknown but minus grad f of xn plus mn plus 1 equals this quantity which is known at time step n and this can be viewed as a noisy estimate of minus grad f of xn. So, in order to estimate lambda we can consider now the following algorithm.

$$-\nabla f(x_n) + M_{n+1} = I_{n+1}(X_n + P) - \frac{\lambda}{\lambda + P} x_n - \frac{\lambda P}{\lambda + P} - \frac{x_n P}{\lambda P} + \frac{\lambda P}{\lambda + P}$$

$$= I_{n+1}(x_n + P) - x_n$$



Online Algorithms for Estimating Change Rates of Web Pages[1]

Konstantin Avrachenkov[a], Kishor Patil[b,*], Gugan Thoppe[b]

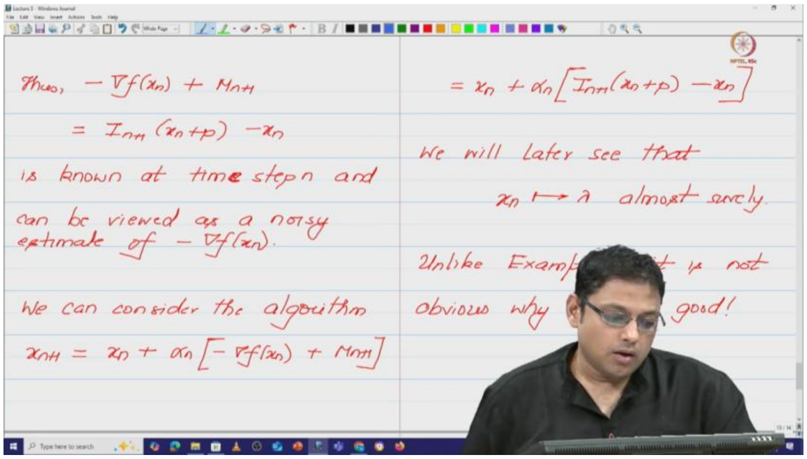
[a]INRIA Sophia Antipolis, 2004 Route des Lucioles, Sophia-Antipolis, 06902, France
[b]Indian Institute of Science, CV Raman Road, Bengaluru, 560012, India

Abstract

A search engine maintains local copies of different web pages to provide quick search results. This local cache is kept up-to-date by a web crawler that frequently visits these different pages to track changes in them. Ideally, the local copy should be updated as soon as a page changes on the web. However, finite bandwidth availability and server restrictions limit how frequently different pages can be crawled. This brings forth the following optimization problem: maximize the freshness of the local cache subject to the crawling frequencies being within prescribed bounds. While tractable algorithms do exist to solve this problem, these either assume the knowledge of exact page change rates or use inefficient methods such as MLE for estimating the same. We address this issue here.

We provide three novel schemes for online estimation of page change rates, all of which have extremely low running times per iteration. The first is based on the law of large numbers and the second on stochastic approximation. The third is an extension of the second and includes a heavy-ball momentum term. All these schemes only need partial information about the page change process, i.e., they only need to know if the page has changed or not since the last crawled instance. Our main theoretical results concern asymptotic convergence and convergence rates of these three schemes. In fact, our work is the first to show convergence of the original stochastic heavy-ball method when neither the gradient nor the noise variance is uniformly bounded. We also provide some numerical experiments (based on real and synthetic data) to demonstrate the superiority of our proposed estimators over existing ones such as MLE. Our algorithms are readily applicable to the synchronization of databases and network inventory management.

Keywords: change rate, web crawling, database synchronization, stochastic approximation, heavy ball, momentum, maximum likelihood, Poisson point process.



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

$= I_{n+1}(x_n+p) - x_n$

is known at time step n and can be viewed as a noisy estimate of $-\nabla f(x_n)$.

We can consider the algorithm

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

$= x_n + \alpha_n[I_{n+1}(x_n+p) - x_n]$

We will later see that

$x_n \longmapsto \lambda$ almost surely.

Unlike Examp it is not obvious why good!

So, let me go over this algorithm. This algorithm is xn plus 1 equals xn plus alpha n times this square bracket minus grad f of xn plus mn plus 1. Indeed, we do not know grad f of xn, but as we have pointed out here, the sum of minus grad f of xn and mn plus 1 is indeed known at the time step n, which corresponds to the true time instance Sn plus 1, right? So, while this is not known, the sum is actually known, right? So, we can actually run this algorithm using just the values of In plus 1, right? And, you know, in that paper that I showed you in the previous slide, we have actually shown that Xn converges to lambda, which is the quantity of interest, almost surely.

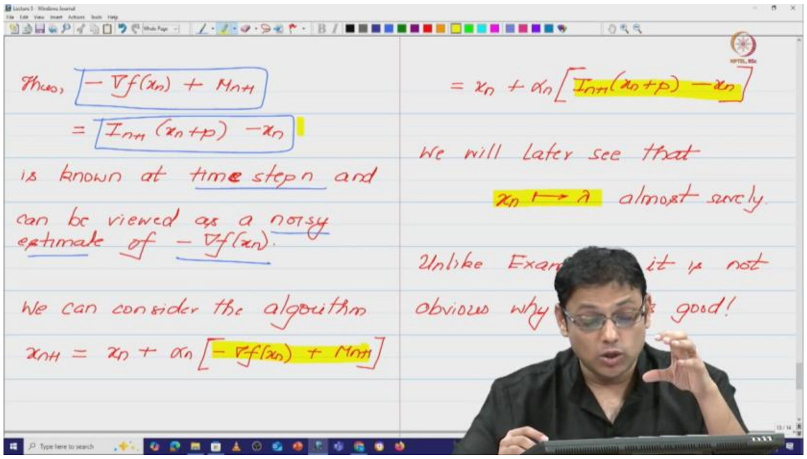
So, the almost surely part comes in here because Xn is a sequence of random variables, and, you know, when we talk of convergence of random variables, we have to use some appropriate quantification. And you will soon see that, you know, when we discuss the convergence of stochastic approximation algorithms, what do we mean by this almost sure convergence and how we can prove this. So, at this point, you know, you only have to note that, you know, this Xn update rule is actually nice. It is nice because, even though you do not know grad f of xn, it is making use of a noisy estimate of grad of xn, which only involves quantities that you know. So, that is what is good about this algorithm on one hand.

In other words, you can actually implement this algorithm. On the other hand, you know, using the theory of stochastic approximation, we will see later that one can show that this Xn actually converges to lambda. So, lambda is the quantity of interest to us, and this algorithm actually converges there, and hence this will be a useful algorithm. Now, one thing that I have not spent too much time on is trying to understand, you know, what is good about this noise sequence or noise term Mn plus 1. I just gave you a hint that, you know, if you take the expectation of In plus 1, then you get this lambda over lambda plus p, something that I mentioned.

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

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

$$= x_n + \alpha_n[I_{n+1}(x_n + P) - x_n]$$



Thus, $\boxed{-\nabla f(x_n) + M_{n+1}}$

$= \boxed{I_{n+1}(x_n+p) - x_n}$

is known at time step n and can be viewed as a noisy estimate of $-\nabla f(x_n)$.

We can consider the algorithm

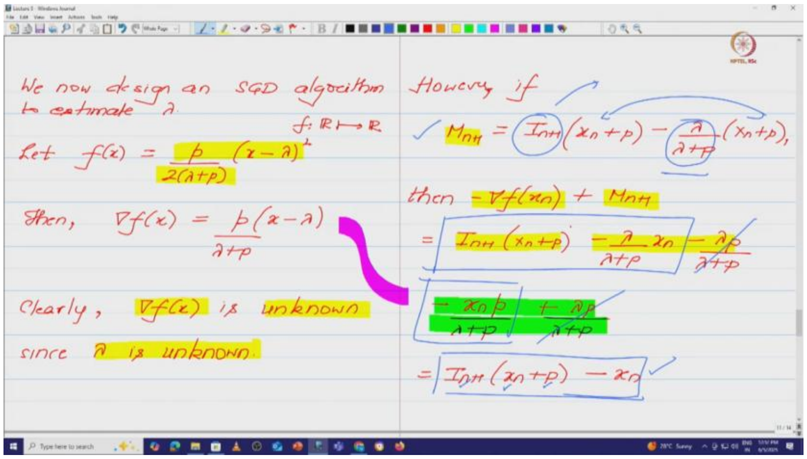
$x_{n+1} = x_n + \alpha_n\left[-\nabla f(x_n) + M_{n+1}\right]$

$= x_n + \alpha_n\left[I_{n+1}(x_n+p) - x_n\right]$

We will later see that

$x_n \mapsto \lambda$ almost surely.

Unlike Exam it is not obvious why good!



We now design an SGD algorithm to estimate λ.

$f: \mathbb{R} \mapsto \mathbb{R}$

Let $f(x) = \frac{p}{2(\lambda+p)}(x-\lambda)^2$

Then, $\nabla f(x) = \frac{p(x-\lambda)}{\lambda+p}$

Clearly, $\nabla f(x)$ is unknown since λ is unknown.

However if

✓ $M_{n+1} = I_{n+1}(x_n+p) - \frac{\lambda}{\lambda+p}(x_n+p)$,

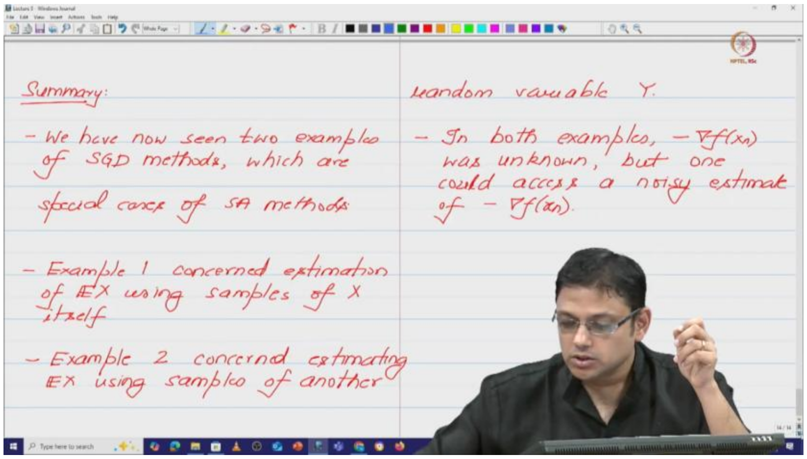
then $-\nabla f(x_n) + M_{n+1}$

$= \boxed{I_{n+1}(x_n+p) - \frac{\lambda}{\lambda+p}x_n} \cancel{- \frac{\lambda p}{\lambda+p}}$

$\boxed{-\frac{x_n p}{\lambda+p}} \cancel{+ \frac{\lambda p}{\lambda+p}}$

$= \boxed{I_{n+1}(x_n+p) - x_n}$ ✓

And then this Xn plus P term is the same in both the first summand and the second summand. So, other than that, I have not told you, but you will soon see that this Mn plus 1 is a nice noise sequence—nice in the sense that you can apply a lot of existing mathematical theory to understand the behavior of this noise term Mn plus 1. So this is something that we will understand in future classes. So now let me finish this lecture 3 by summarizing what we have done so far. So far, we have seen two examples of SGD methods.



Summary:

- We have now seen two examples of SGD methods, which are special cases of SA methods

- Example 1 concerned estimation of $\mathbb{E}X$ using samples of X itself

- Example 2 concernd estimating $\mathbb{E}X$ using samples of another random variable Y.

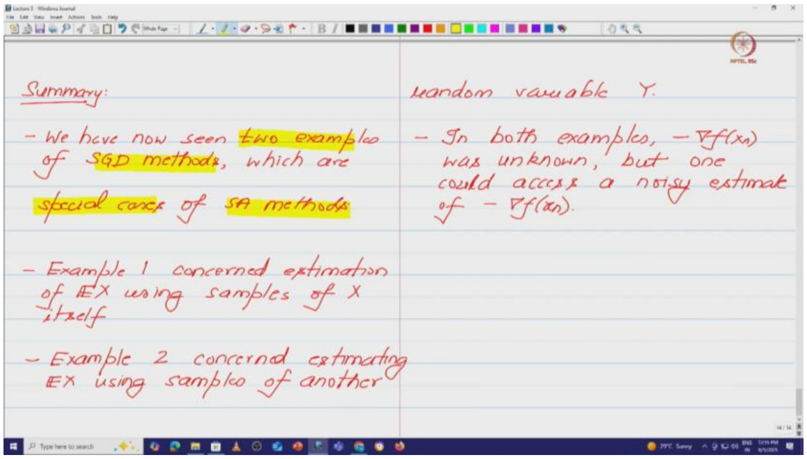
- In both examples, $-\nabla f(x_n)$ was unknown, but one could access a noisy estimate of $-\nabla f(x_n)$.

So I think I did not emphasize it. Even for example 2, the algorithm that we came up with is actually an SGD method—that is, stochastic gradient descent—and as I have mentioned in the first class, SGD methods are special cases of stochastic approximation methods. Now, we have looked at two examples. In example 1, we looked at the case of estimating the expected value of X using samples of the random variable—random vector X itself. In example 2, we looked at estimating the expectation of a random variable X.

We wanted to estimate this quantity lambda, which, in some sense, the inverse of this was the expected value of the interarrival times—that is, tn plus 1 minus tn—and we used this, I mean, we did this estimation using samples of another random variable y. In this case, the random variable y that we had access to were these in plus 1, i1, i2, i3, and so on. So, we made use of these values to estimate something about lambda. And we also noted that in the two examples, the true gradient was inaccessible—it was unknown because it depended on quantities that were unknown. But one could access a noisy estimate of these negative gradients.

And because we could make use of, you know, these noisy estimates, we can actually run these SGD methods in practice. Even though we do not know these unknown quantities, in particular, we do not know these unknown gradients. We can actually run this algorithm, and in future lectures, we will see how we can analyze these algorithms. In particular, we will see how we can understand the convergence of these algorithms. That is one thing we will discuss in a future class. In the next class, we will also try to see an example of a stochastic approximation method that is not an SGD.

So, as I told you, SGD methods are a special subclass of stochastic approximation methods. However, the true power or potential of stochastic approximation lies in the fact that we can also view them beyond the standard SGD methods. Okay, with this, let me stop. Thank you, and namaste!



Summary:

- We have now seen two examples of SGD methods, which are special cases of SA methods
- Example 1 concerned estimation of $\mathbb{E}X$ using samples of X itself
- Example 2 concerned estimating $\mathbb{E}X$ using samples of another random variable Y.
- In both examples, $-\nabla f(x_n)$ was unknown, but one could access a noisy estimate of $-\nabla f(x_n)$.