STOCHASTIC APPROXIMATION: THEORY AND APPLICATIONS

Dr. Gugan Thope

Department of Computer Science and Automation

Indian Institute of Science

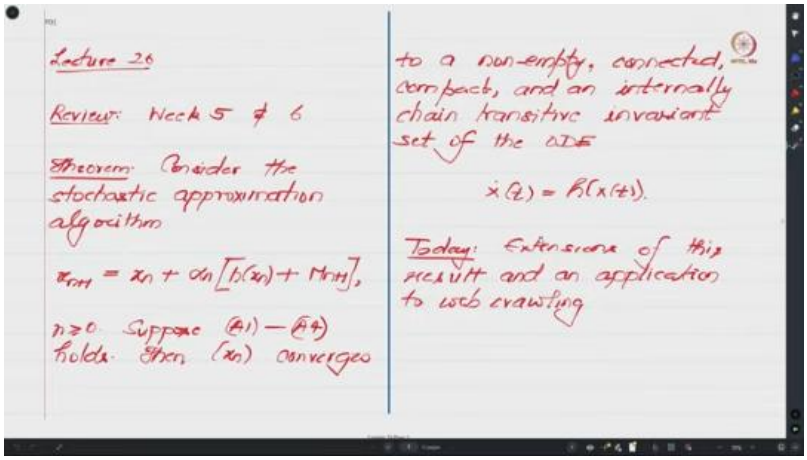
Week 6

Lecture 26

Extensions, Variants, and Applications of Stochastic Approximation

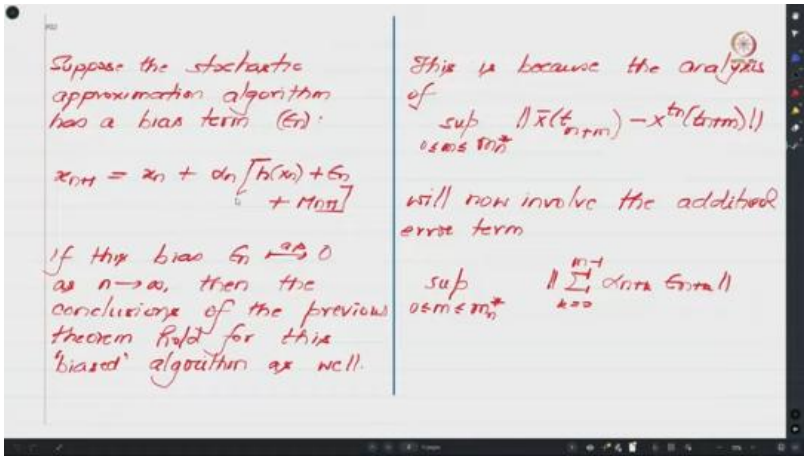
Hello and Namaste everyone. Welcome to lecture 26 of this NPTEL course on Stochastic Approximation. This will be the last class of week 6. So, let us do a quick recap of what we have been doing so far, and then we will give an overview of what we are planning to do today. In this week, we basically discussed some sufficient conditions under which we can study the asymptotic behavior of a stochastic approximation algorithm in terms of invariant sets of its limiting ODE, right?

So, what we will do in this class is discuss some generalizations of this result, and thereafter we will see how we can verify some of these assumptions for the algorithm—in particular, one algorithm that we had discussed during the first week of this course. So, let us begin with the details. So, what we have been doing so far can be summarized over here. So, we have proved this theorem, which states that suppose you have a stochastic approximation algorithm which has an update rule of the following form. Furthermore, suppose a bunch of assumptions hold; then the iterate sequence generated by this algorithm converges to a non-empty, connected, compact and internally chain transitive invariant set of the ODE x dot of t equals h of x of t, right?



And as I told you, these characterizations help in restricting the possible choices for the invariant sets to which an algorithm can converge. For example, if your stochastic approximation iterates live in the R^D space, then trivially R^D is an invariant set. However, if you insist that it should be compact, then that trivial invariant set is ruled out. So, by having these finer characterizations, we can limit the choices for the limiting sets. So, what we are going to do today is discuss some extensions of this result. In particular, we will discuss a case where this algorithm is slightly different from this, and then we will discuss a generalization of this result where we have an existence of what is called a Lyapunov function, right? And then, at the end of this class, we will discuss a specific

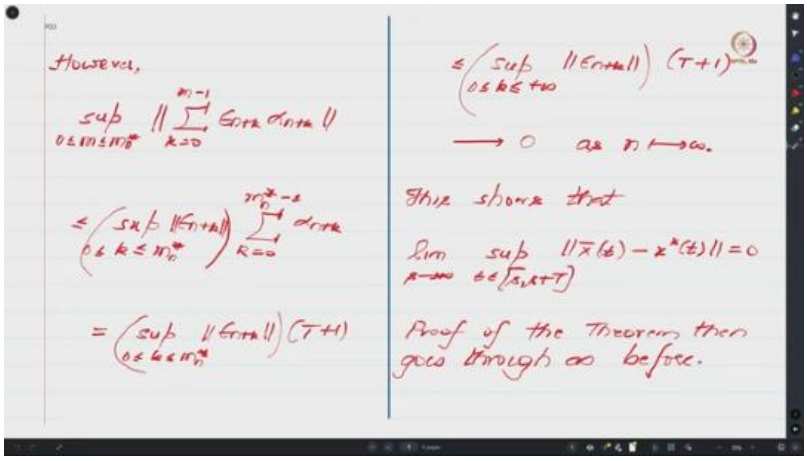
Algorithm that we introduced in the first week and show you how one can verify these assumptions A1 to A4 in that context. So, the first extension that we will talk about is the algorithm of the following form. So, notice that the difference between this algorithm and the one we introduced on the first slide is that this new algorithm has the presence of this bias term epsilon n. So, the algorithm, instead of having the form H of Xn plus Mn plus 1, we are saying that the algorithm has the form H of Xn plus Mn plus 1 plus an additional perturbation term epsilon n. Which we will denote as bias.



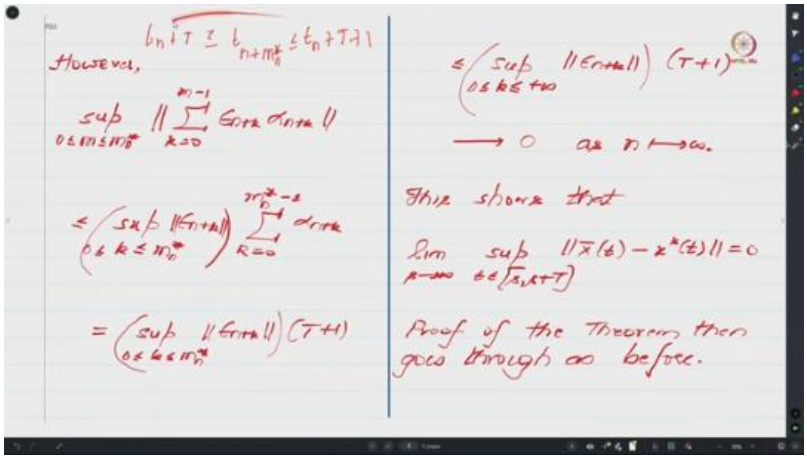
$$x_{n+1} = x_n + \alpha_n[h(x_n) + \epsilon_n + M_{n+1}]$$

So, the way to interpret this is that, ideally, we would have liked something like this, but because of the presence of noise, we can only get a noisy estimate of H. What we are saying here is that instead of having zero-mean noise, we have a biased estimate of H, and in fact, in several practical scenarios, this epsilon n often will be non-zero. So, what this extended result says is that, suppose for some reason this epsilon n term goes to 0 as n tends to infinity, then the conclusions of the theorem we discussed in the previous slide will hold for the biased algorithm as well. And the reason is that Whatever analysis we had done in the previous few classes, where we looked at bounding the distance between X bar TN plus M, which is the linear interpolation of your XN iterates, and XTN TN plus 1, which is the solution trajectory of your limiting ODE. So, whatever analysis we had done to bound an expression like this will now only involve an additional term of the following form.

So, notice that there is this alpha n plus k epsilon n plus k, which comes by the product of these kinds of terms. Right, and the analysis will now involve this additional term. Now, I will quickly show you why this additional term is not going to cause harm as long as this perturbation bias goes to 0 in an asymptotic sense. In particular, if you look at the supremum over the sum of k equals 0 to m minus 1 of epsilon n plus k alpha n plus k, so this is like the cumulative effect of the perturbation bias right from time n all the way up till time n plus m. So, this is the cumulative effect, and now we are looking at the supremum of the norm of this expression for m, which lies between 0 to mn. So, this notation Mn star I had actually defined in the previous class. So, you may want to check it up.



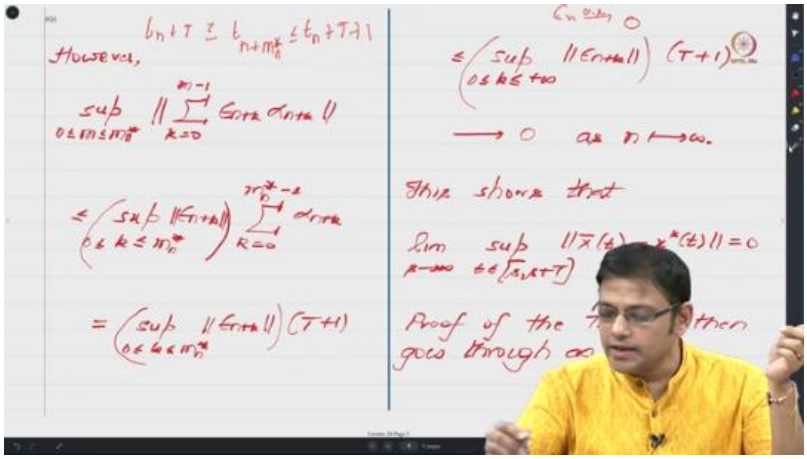
But the idea there was that this Mn star ensures that your Tn plus Mn star is something that satisfies a relation of the following form. So, in some sense, we are now able to cover all the indices that are there between Tn and Tn plus T plus 1. So, if you remember a window of length T plus 1. So, that is what we are looking at over here, and by a simple bound, one can see that if you take the supremum of the norms outside, we will be left with something like this. Right, and wherever there was M, we have replaced it with Mn star because we are taking the supremum.



So, the larger supremum or a supremum over a larger set is always bigger, and similarly here we had M. So, if you take the sum of these step sizes from k equals 0, the supremum will occur when you replace M with Mn star, right? So, you will have something like this, and since we are satisfying this condition over here, one can check that these step sizes The sum of these step sizes is actually upper bounded by T plus 1. Is that okay?

So, eventually, we have managed to show that this expression over here, right? This expression over here is upper bounded by a term of this form, right? And I can now drop this Mn star and replace it with plus infinity. So, again, I am taking a supremum over a larger set. Hence, this supremum will be larger than this.

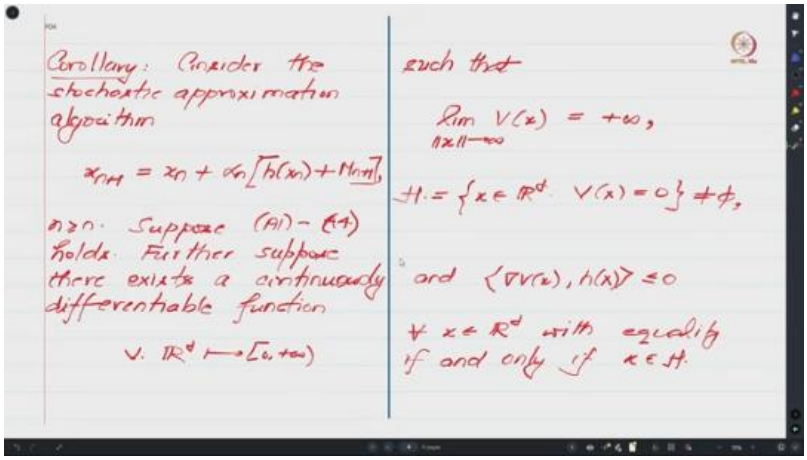
And again, observe that since k goes from 0 to infinity, the first term here is epsilon n, the subsequent term here is epsilon n plus 1, and so on and so forth. So, this is a constant, and this expression will go to 0 almost surely as n tends to infinity because of our assumption that epsilon n goes to 0 almost surely. So, this is something that we have assumed. And once we show something like this, one can immediately show that this limit goes to 0, which was the conclusion of the technical lemma that we proved. And once this lemma is established, the proof of the theorem goes through as before because the proof of the theorem only relied on a result of the following form.



So, in this sense, the conclusions of the theorem also apply to the algorithm involving this biased term. So, now we are going to discuss a corollary to our main result. So, for this case, we go back to the stochastic approximation algorithm without the biased term. This is just to keep the discussion simple.

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

This corollary can also be extended to the case involving the epsilon n term.



So, what does this corollary say? It says that suppose you have a stochastic approximation algorithm of the following form, right? And suppose assumptions A1 to A4 hold. Further, suppose there exists a Lyapunov function, which means that there exists a continuously differentiable function V whose domain is Rd and whose range of values is from 0 to infinity. And let us suppose this function V satisfies the following properties, which is what makes it a Lyapunov function.

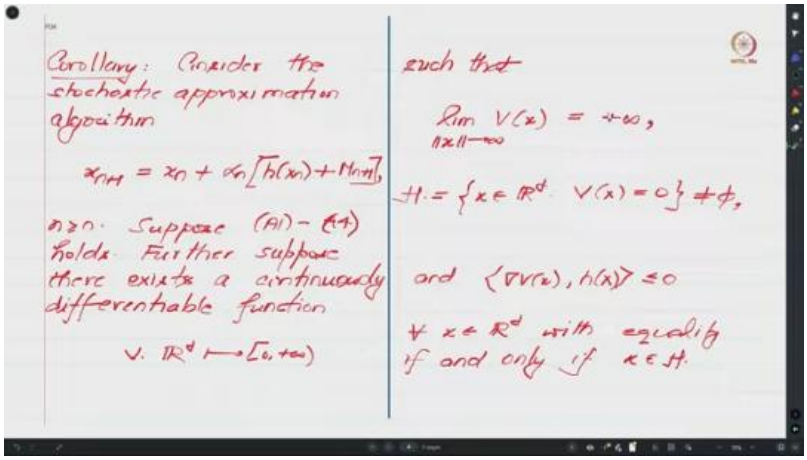
The first of these properties we require is that the value of v of x, so keep in mind that for x, I mean x is an element in Rd, v of x is a non-negative real number and we require that as the norm of x goes to infinity, the value of v of x also grows to infinity, that is As X becomes larger and larger, we require that V of X also take sufficient, I mean like the value that V of X takes also grows to infinity, right?

$$V: \mathbb{R}^d \mapsto [0, +\infty)$$

$$\lim_{||x|| \to \infty} V(x) = +\infty,$$

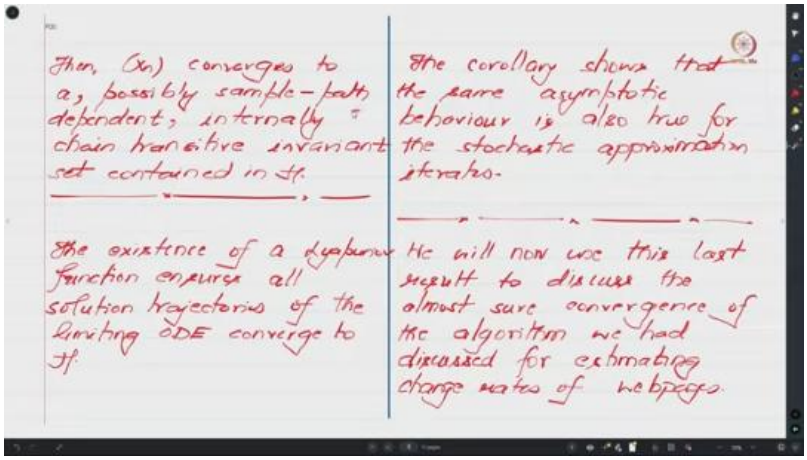
$$H: \{x \in \mathbb{R}^d : V(x) = 0\} \neq \phi$$

And the set H of values where V of X is 0 on one hand is non-empty. On the other hand, we require that the inner product between the gradient of V of X and H of X. So, keep in mind that V is a function from Rd to the set of non-negative real numbers.



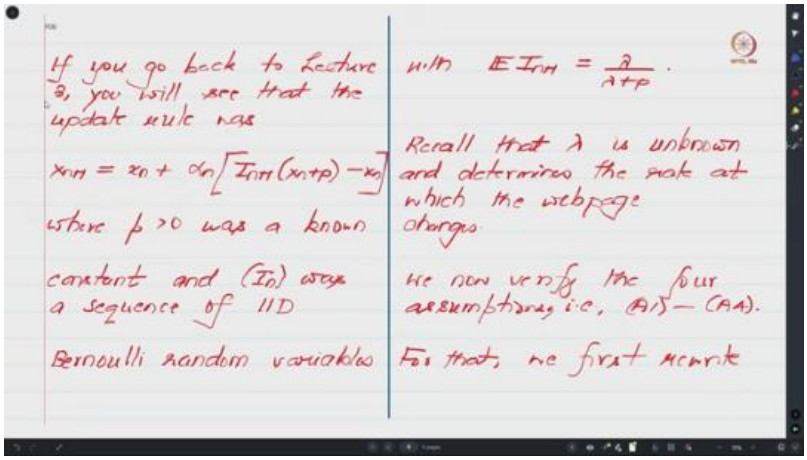
Hence, its gradient will be d-dimensional. h of x again is d-dimensional. Hence, we are talking of an inner product here. So, we require that this inner product be non-negative with equality if and only if x belongs to h. So, let me go over the conditions one more time. So, a function V is said to be a Lyapunov function if it satisfies these properties that is that as the norm of X goes to infinity, V of X blows up to infinity.

The set H of values where V of X is 0 is non-empty and it is also the place where this gradient is taking, I mean this inner product takes the value 0 and for every X which is not in H, this inner product is strictly less than 0. So, whenever we have such a V we will say that we have a global Lyapunov function right. So, what this corollary says is that if you have a stochastic approximation algorithm if these assumptions hold and there exists a Lyapunov function. Then Xn converges to a possibly sample path dependent internally chain transitive invariant set contained in H. So, here if you observe we are able to make a stronger conclusion that is we are saying that you know the limiting invariant set to which your stochastic approximation iterates could converge to they will lie within H. So, once we identify H, we can further narrow down the possible choices for the invariant sets to which your algorithm can converge to, right.



And, you know, now in the rest of today's discussion, we are going to discuss an example where, you know, we will sort of touch upon all these ideas that we have just discussed. So, I would like to add one point before we move ahead. So, you know, what does the presence of a Lyapunov function imply? So, the first thing we can observe is that, I mean, this is from the literature on differential equations that— If you have a Lyapunov function, then one can show that all solution trajectories of the limiting ODE will converge to H. This is something we can verify.

This is because of this condition over here, which implies that the value of V along the solution trajectory of the limiting ODE is decreasing and hence it will stop only when or converge to a place where the value of V is 0, which is the smallest possible value that we can take. So, the existence of a Lyapunov function ensures that the solution trajectories of the limiting ODE converge to H. What this corollary tells us is that the same asymptotic behavior also holds true for the stochastic approximation iterates. So, now, as I said, in the rest of today's discussion, I am going to use this last result to discuss the almost sure convergence of one of the algorithms that we had discussed in the first week—in particular, the algorithm that we had used for estimating the change rates of the web pages. In particular, I had discussed this in Lecture 3. So, if you go back, you will see that for this algorithm, the update rule was of the following form.

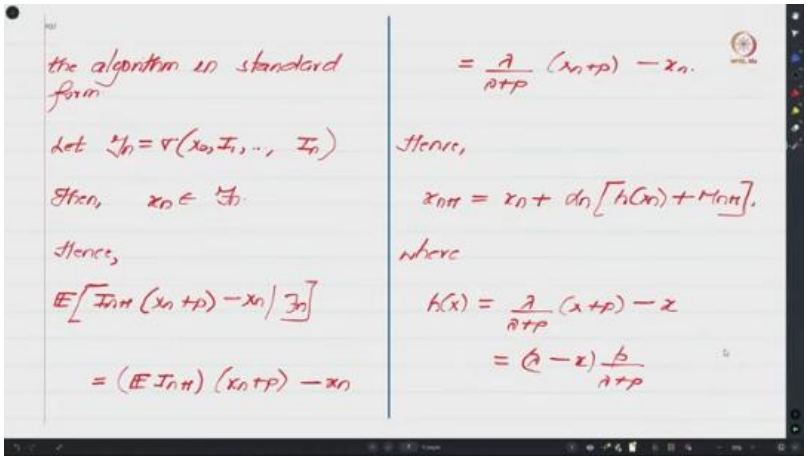


That is xn plus 1 equals xn plus alpha n times in plus 1 times xn plus p minus xn. So, I will not have time to go over the details of all the different things. For all the details and, you know, why we came up with this algorithm and so on. I request you to go through Lecture 3. Here, I am going to discuss the convergence analysis of this algorithm, right?

But you may want to keep in mind that the P we have over here, okay, is a known constant. So, we do not have to worry about it. And the sequence I_n is These are, you know, independent and identically distributed Bernoulli random variables—that is, I_{n+1} takes values 0 and 1. with their expectation being $\lambda/(\lambda + P)$, where λ is unknown and determines the rate at which a particular web page changes.

So, in some sense, this λ is what we want to compute, right? And this algorithm was proposed as one of the ways in which this λ can be estimated. Is that okay? So, now P is known, these I_ns are IID Bernoulli random variables with their expectation being $\lambda/(\lambda + P)$, where λ is unknown. And now what we are going to do is, for this algorithm, we are going to verify the four assumptions A1 to A4 and then use the conclusion of the theorem to show or figure out where this algorithm will asymptotically converge to.

Right. And before we start verifying these assumptions, what we will do is we will first rewrite this algorithm in the standard stochastic approximation form. That is, we want to write it in the form $X_{n+1} = X_n + \alpha_n H(X_n) + m_{n+1}$. So, how can we do that? For that, first let us define the filtration of sigma fields where F_n is the sigma field generated by these quantities.



That is the initial estimate X0 and the values of I1 to In. And one can easily see that your Xn that is the estimate at the nth time instance that is a function of these variables. So, hence if you know these values Xn is completely known and hence one can formally conclude that Xn is actually measurable with respect to Fnp. So, once we have this sigma field and establish that Xn is measurable with respect to Fn, what we do is whatever is the update rule over here, we take its conditional expectation with respect to this sigma field. So, if you take the conditional expectation of this with respect to this sigma field, we have just shown that Xn is measurable with respect to Fn.

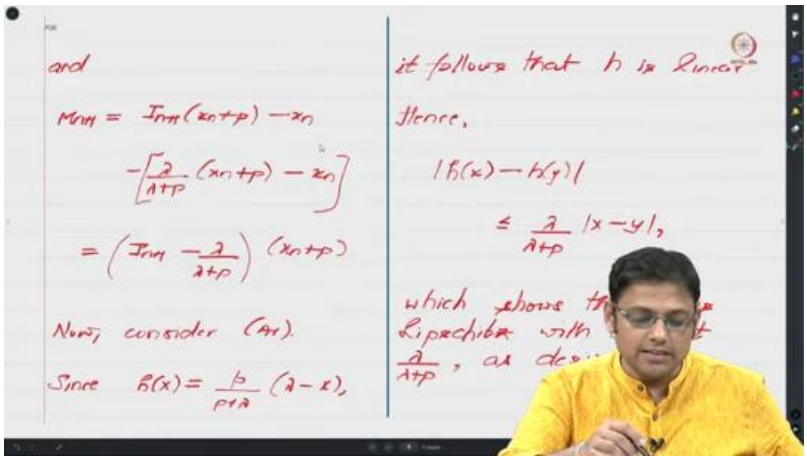
Hence, in this update rule, this whatever is there that depends on Xn will actually come out of the conditional expectation as is. And we also know that these INs are independent of the past. Hence, this conditional expectation will not apply to it. And what we will end up with is the standard expectation, right? So, this conditional expectation will turn out to be expectation of IN plus 1 times XN plus P minus XN.

And we know that expected value of i n plus 1 is lambda over lambda plus p. This is something that we had established, you know, in lecture 3 itself, right? So, I request you to go to that lecture to see why that is the case. Hence, one can, you know, now keeping this observation in mind, rewrite this algorithm in the following way, where h of x n is is basically whatever you have over here right. So, if you observe whatever you have over here that is exactly what I have used to define h of x I have just replaced x n with x right.

So, you end up with some expression like this, and one can see that, you know, this can be simplified further to this. One can see that lambda times x. cancels with this lambda times

this x, and you have lambda times p minus x times p. So, you can take the p common, and you will end up with lambda minus x times p over lambda plus p. Is this okay? So, of course, this h of x is unknown because lambda is unknown. So, I am not claiming that you can compute h of xn, but rather what we are claiming is that you can compute or estimate h of xn plus mn plus 1 because this expression is precisely what you have over here. At time n, you will know xn, you will know p, and the value of in plus 1 will also become available, whether it is 0 or 1. It will become available at time instance n.

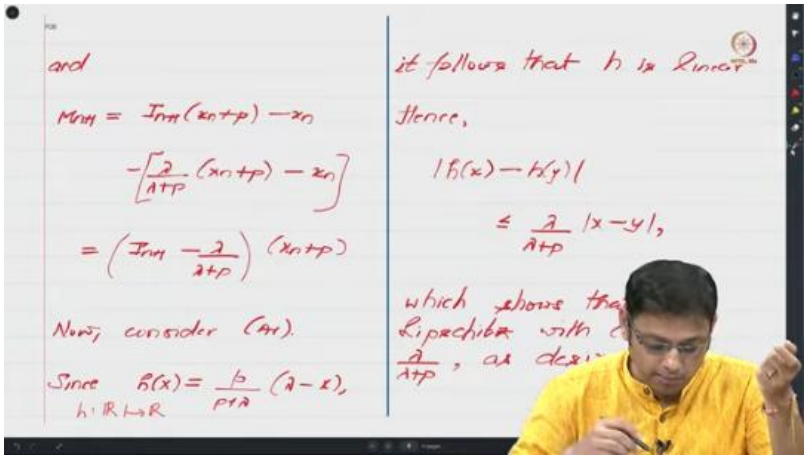
So, since we have now added H of Xn here, Mn plus 1 is basically whatever we had before minus its conditional expectation, right? So, one can see that if you take H of Xn and add Mn plus 1 to it, this part will cancel off, and we will be left with what we had in the original algorithm, right? And again, if you simplify this a bit more, you can see that this expression actually translates to In plus 1 minus lambda over lambda plus p times Xn plus p. I have basically cancelled these parts over here, right? And hence, you will end up with this. So, now we have the expression for H and the expression for Mn plus 1.



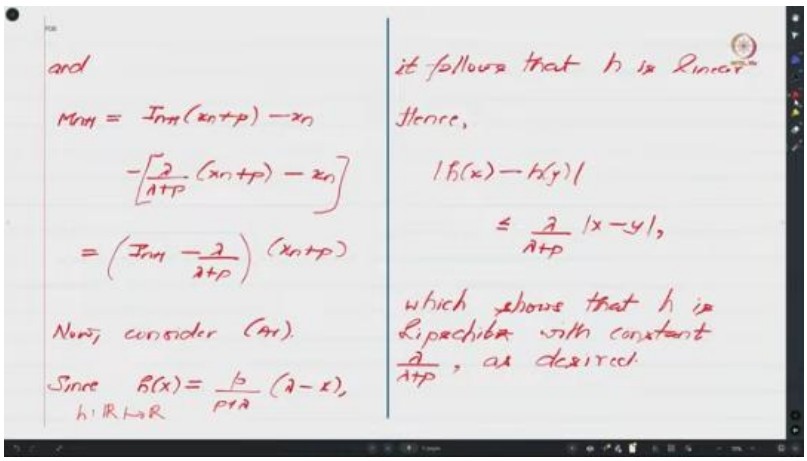
And what we are going to do now is to verify these four assumptions that we, you know, stated in the first lecture of week 5, right? So, let us verify assumption A1. In assumption A1, we need to check whether this function H is Lipschitz continuous or not. So, towards that, observe that H of X has this form over here. And in particular, you see that this is a linear term.

So, I would like to highlight here that H is a function from R to R. Is this okay? So, it takes in a real number as input and outputs a real number, right? And one can see that because

H of X has this form over here, one can conclude that it is linear. I mean, formally one should say that it is an affine function. Right.

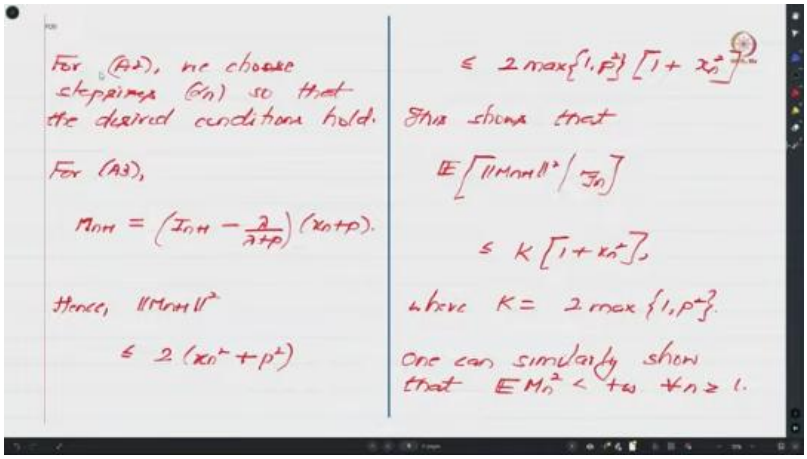


And one can then see that if you look at the distance between H of X and H of Y. So, notice that I look at absolute values here, the reason being that H is a real-valued function. So, if you look at the absolute value of H of X minus H of Y, right, one can easily see that I think I have made a mistake here, this should be P over lambda plus P times X minus Y. So, this constant term will cancel off both in H of X and H of Y, and we will only have this term, hence we will have P over lambda plus P times the absolute value of X minus Y. So, this precisely shows that H is Lipschitz continuous with Lipschitz constant. So, again I have made a mistake here. This should be P over lambda plus P.

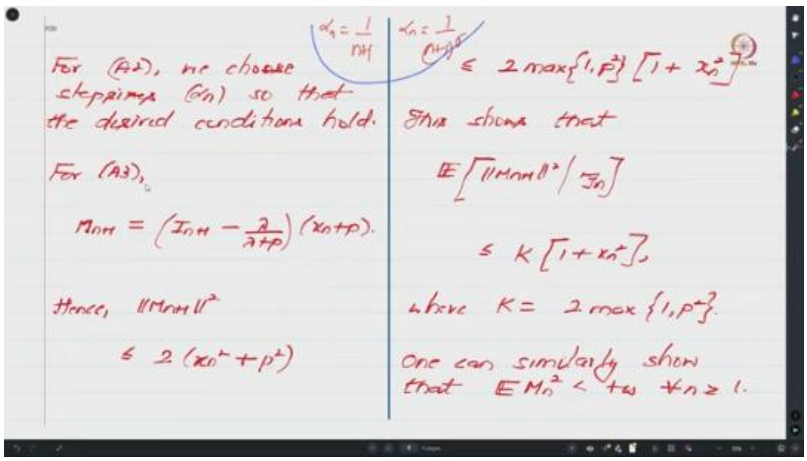


So, this shows that H is Lipschitz continuous with Lipschitz constant P over lambda plus P. And which establishes A1 as desired. Now, with regards to A2, we will basically choose

a step size sequence which satisfies the desired condition, which is that your summation alpha n should be infinity while summation alpha n squared should be less than infinity. So, I have already told you examples of such step sizes; for example, one could choose alpha n to be 1 over n. Or alpha n equals 1 over n plus 1 raised to sigma.



I mean, several variants of these step sizes also exist. You know, you can either figure it out yourselves or you can look at textbooks, right. But anyways, these step sizes are quite standard, and one can see that once you pick a step size of this form, where sigma is strictly bigger than half, then indeed your A2 assumption holds. So, now let us focus on A3. So, in A3, we need to show a bunch of things. First is that Mn plus 1 is a martingale difference sequence.



Now, the way we had defined Mn plus 1. So, if you go back over here, Mn plus 1's definition is you take this expression and from it subtract the conditional expectation of

this expression. So, the way we have defined it, one can trivially see that your Mn plus 1 indeed will have the martingale difference property, which is that if you take the conditional expectation of Mn plus 1 with respect to the information that you have at time n, which is fn, you will see that this conditional expectation will indeed be 0, which tells us that the sequence of Mn values is indeed a martingale difference sequence. The next thing that we had to verify under A3 was that your Mn is actually 0.

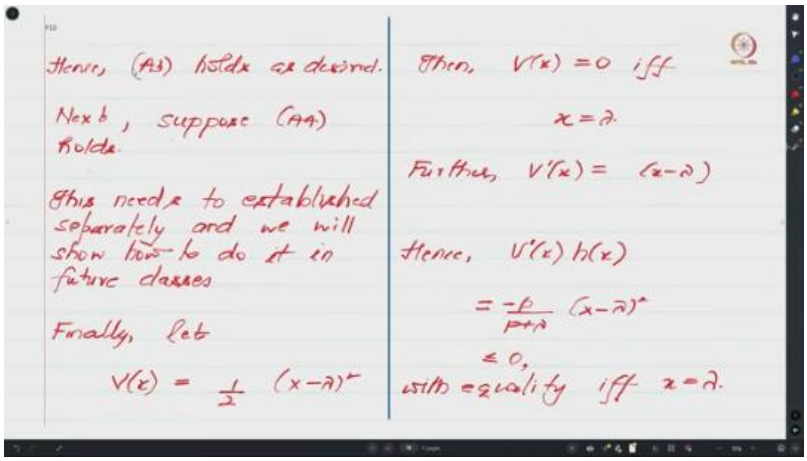
So, you know, I mean, the Mn sequence is in L2, which requires us to check that the expected value of Mn squared is less than infinity for all n, right? And this bound on the expected value of Mn squared can vary from one n to the other. Right. And I encourage you as a reader to verify that. And that is very straightforward. In fact, you know, you can show that, based on this update rule that you have over here, one can show that Xn cannot grow very fast.

So, Xn can grow. So, you can sort of come up with a worst-case bound for norm Xn or absolute value of Xn. And then use that to show that Mn plus 1 will also be bounded for each n, with a bound that grows with n, but there is a bound. And once you have a bound, it immediately shows that the expected value of Mn square is less than infinity. So, that is also something that is straightforward to verify.

The last thing that we need to verify is to show that there exists some constant K such that the conditional expectation of—so here, I should perhaps say—the absolute value of Mn plus 1 square is less than K times 1 plus Xn square. So, again, this is very easy to verify. So, one can, for example, see that Mn plus 1 square is actually less than the absolute value of this times the square of this, but the absolute value of this is upper-bounded by 1. So, let me write this down. So, because your In plus 1 is a number,

which is either 0 or 1, and lambda over lambda plus p again is a number between 0 and 1. So, one can see that this value is trivially upper-bounded by 1. And the square of this expression is upper-bounded by 2 times Xn square plus p square. And hence, one can see that Mn square is upper-bounded by 2 times Xn square plus p square. By taking this P out in common, one can show that this expression is upper-bounded by 2 times the max between 1 and P square times 1 plus Xn square.

And from this, one can trivially conclude that since your Mn square itself is upper-bounded by some constant times 1 plus Xn square, if you take the conditional expectation of this, then indeed one can show that this is true for K equals 2 times the max of 1 comma P square. Is this okay? And with this, we have actually verified that A3 also holds, right? Now, what remains is to verify A4. Now, to verify A4, we require some additional analysis, which we will be doing in one of the future classes.



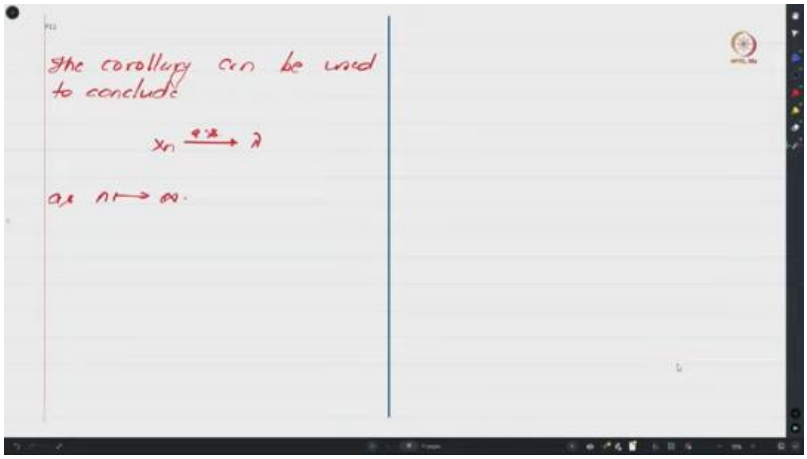
So, wait until then, right? So, at this point, just presume that A4 holds. We will separately show that in one of the previous—I mean, future weeks, when we will talk about the stability of stochastic approximation algorithms. So, let us say A4 holds, and now I would like to show that suppose we define a V function in this way, that is, V of X is half times X minus lambda square.

I will show that this V function is Lyapunov, and importantly, the only zero of this V function is lambda itself. So, if I can somehow show that V is a Lyapunov function, then from that corollary, I can establish that since A1 to A4 assumptions hold, your Xn will almost surely converge to lambda, which was the parameter that we were wanting to estimate. Right. And the verification of V being a Lyapunov function is straightforward. Observe that V of X, first of all, is 0 if and only if X is lambda.

So, this IFF stands for if and only if. So, V of X equals 0 if and only if X equals lambda. And if you take the derivative of V. So, here I do not write nabla V of X because this is a real-valued function. So, V prime of X is trivially X minus lambda, and now if you go back

and check the definition of H and multiply it with V prime X, one can see that we end up with the expression minus P over P plus lambda times X minus lambda the whole square.

Right, which is indeed non-negative with equality if and only if X equals lambda. So, this completes our verification of V being a Lyapunov function. And we can now conclude that, you know, all the assumptions of this corollary hold, right. Of course, this A4 is something that we will verify in the future, but modulo that. All the assumptions hold, and hence one can in fact conclude that Xn converges almost surely to lambda.



In other words, that simple algorithm that we had seen over here. Actually converges almost surely to lambda, which means that every time you run this algorithm asymptotically, it will converge to lambda, which was the unknown quantity that we were wanting to estimate. So, this brings us to the end of today's class. Let me give a quick summary of what we have done in week 5 and week 6. In weeks 5 and 6, we discussed the analysis of asymptotic convergence of stochastic approximation algorithms.

In particular, we gave these four sufficient conditions concerning the H function, concerning the step sizes, concerning the noise, and also, we had assumed in A4 that the iterates are almost surely bounded. So, this last assumption A4 needs to be verified separately, and we will do it in one of the upcoming weeks. But once we establish these four assumptions, we showed that your stochastic approximation algorithms indeed converge to suitable invariant sets of your limiting ODE. In particular, in today's class, we discussed some extensions of this result and also considered an example from the world of web crawling, showing that for this algorithm we have at hand for estimating lambda.

Indeed, it satisfies those A1 to A3, and it will also satisfy A4, but we will establish that in a future week. Since it satisfies these and because there exists a Lyapunov function, we were able to conclude that this algorithm almost surely converges to lambda, which determines the rate at which the web page changes and was the quantity of interest.

So, next week, we will discuss convergence rates of stochastic approximation algorithms. Hopefully, we will join again. Until then, thank you. Goodbye and Namaste.