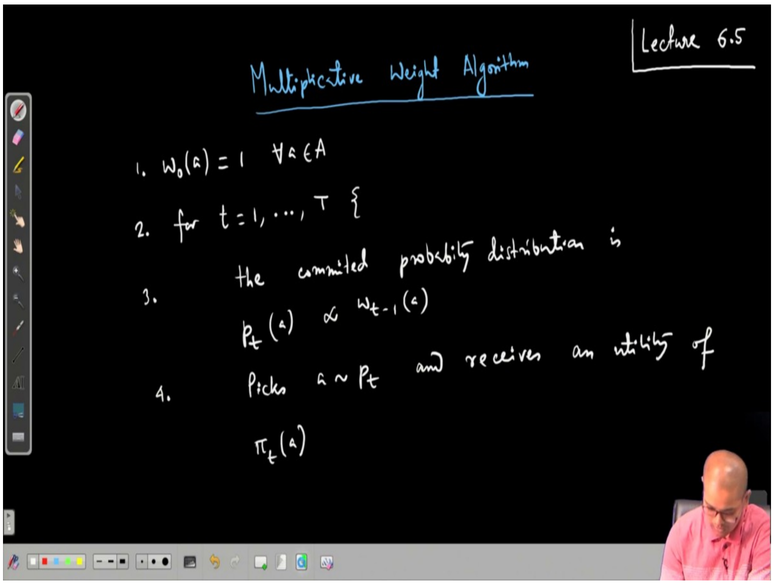
Algorithmic Game Theory
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 30
Multiplicative Weight Algorithm

Welcome. So, in the last class we have started studying No-Regret Dynamics and we have mentioned that there is a no regret algorithm, no external regret algorithm which is called Multiplicative Weight Algorithm and this we will see in today's lecture.

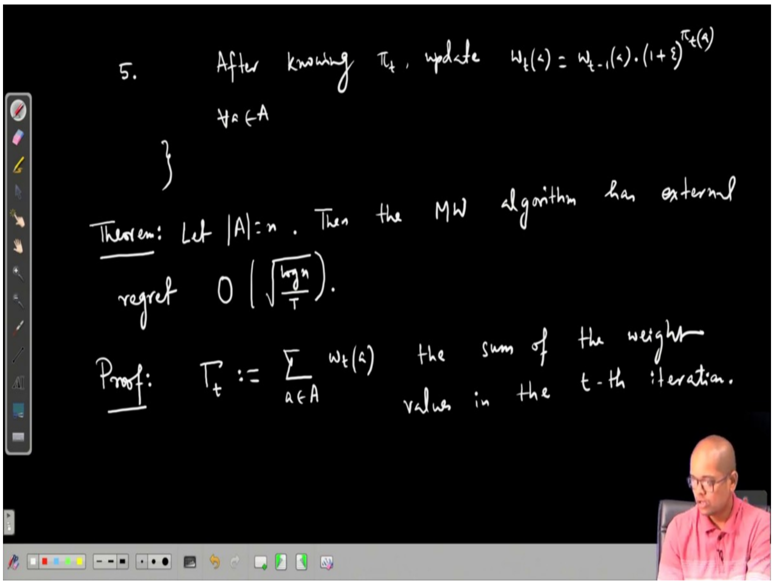
(Refer Slide Time: 00:45)



This lecture Multiplicative Weight Algorithm. So, let us first describe the algorithm. So, step 1, it maintains a vector of weights which in every iteration it updates. So, it initialized to 1. So, $w_0(a)=1$ for all action $a \in A$, then for every iteration t equal to 1 to capital T. So, it the committed probability distribution is $p_t(a)=w_{t-1}(a)$ by it is just probabilities are proportional to this weights. So, I just need to normalize it or let me write proportional ok.

So, then the adversary picks utility function π_t. So, there is nothing to do for the algorithm. So, it next picks an action a from this probability distribution p_t and receives an utility of $\pi_t(a)$.

(Refer Slide Time: 03:57)



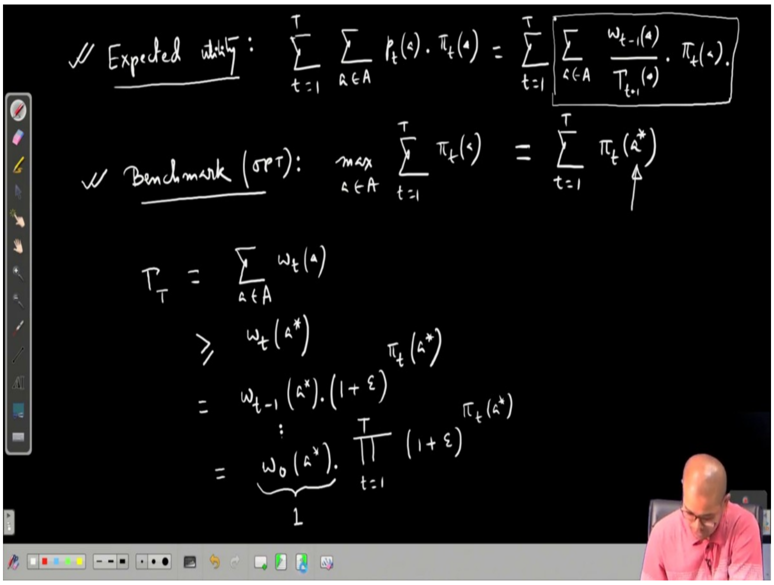
Then it comes to know the utility function π_t and then it updates the rule updates the weights. So, after knowing after knowing π_t update $w_t(a)=w_{t-1}(a)(1+\epsilon)^{\pi_t(a)}$ ok. So, epsilon is a parameter we will choose later these for all action $a \in A$ and that is the end of for loop. So, is the algorithm clear?

So, in every iteration it just picks up probability distribution which assigns probabilities proportional to the weights, it maintains a weight vector which is initialized to all 1 and then after knowing the utility function π_t it updates the weight function and it defines $w_t(a)$, the weight of a to be its existing weight $w_{t-1}(a)(1+\epsilon)^{\pi_t}$. So, the more utility the particular action has in the current iteration it gets more weightage ok.

So, now, let us see. So, what is the main theorem that we want to prove. Theorem: let us recall, let A equal to n size of A equal to n; that means, player has n actions. Then the multiplicative weight algorithm has external regret $O\left(\sqrt{\log\frac{n}{T}}\right)$ ok proof.

So, towards that let us define first the probability vector the p_t's are proportional to the weights, so and how do we define? We define we divide each weight by its normalization factor; that means, sum of the weights. So, what is the normalization factor? Let us call that Γ_t that is nothing but sum of the weight values ok is nothing but the sum of the weight values in the t-th iteration ok, good.

(Refer Slide Time: 08:39)



Now, what is the expected utility? So, a expected utility let us call this expected payoff or expected utility, expected utility is nothing but $\sum_{t=1}^{T} \sum_{a \in A} p_t(a) \pi_t(a)$ this is nothing but $\sum_{t=1}^{T} \sum_{a \in A} \frac{w_{t-1}}{\Gamma_{t-1}} \pi_t(a)$. So, this we will see later.

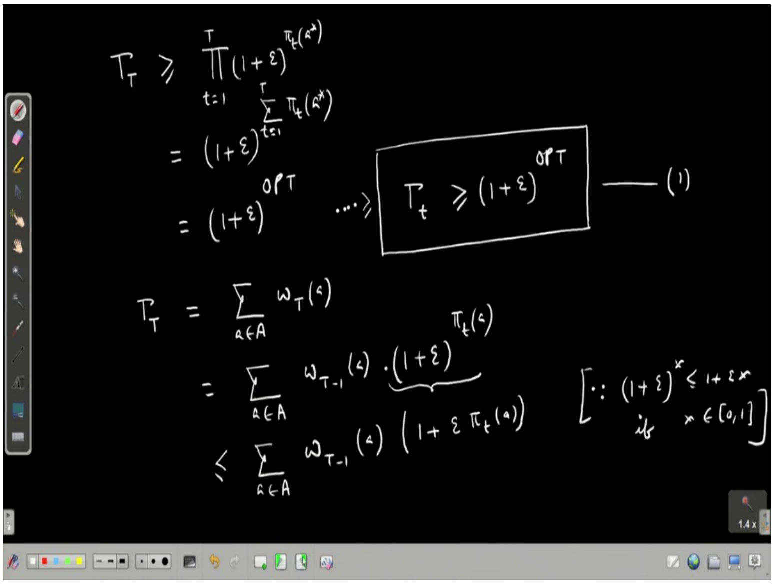
And what is the benchmark? Let us call it OPT. What is OPT? This is $max_{a \in A} \sum_{t=1}^{T} \pi_t(a)$. So, go over all actions and see which action maximizes total utility and that utility is my benchmark. So, suppose that this is a star. So, suppose the action that maximizes total utility is a star. So, this is then $\pi_t(a^*)$ t equal to 1 to capital T ok.

Now, let us see first we need to. So, what is the thing we need to show, we need to relate this extra expected utility with the benchmark and the idea is that we will relate both expected utility and benchmark with Γ_t. So, let us see how. So, first what is Γ_t is by definition $\sum_{a \in A} w_t(a)$ the sum of the weights of all the actions in the t-th iteration that is gamma T.

Now, suppose we want to relate this with the bench mark. So, in the bench mark there is a star here. So, I want to bring a^* and the easiest way to bring a^* is that a^* is one of a and so this is $w_t(a^*)$, weights are all non-negative numbers and that is how we got it. Now, we expand it, what is $w_t(a^*)$, how it is updated? It is $w_{t-1}(a^*)(1+\epsilon)^{\pi_t(a^*)}$ that is just how the algorithms updates this weights.

And we continue this, if we continue this then what we get is that this is nothing but $w_0(a^*)\prod_{t=1}^{T}(1+\epsilon)^{\pi_t(a^*)}$. So, let us continue. What is $w_0(a^*)$? $w_0(a^*)=1$. So, you can forget it.

(Refer Slide Time: 13:38)

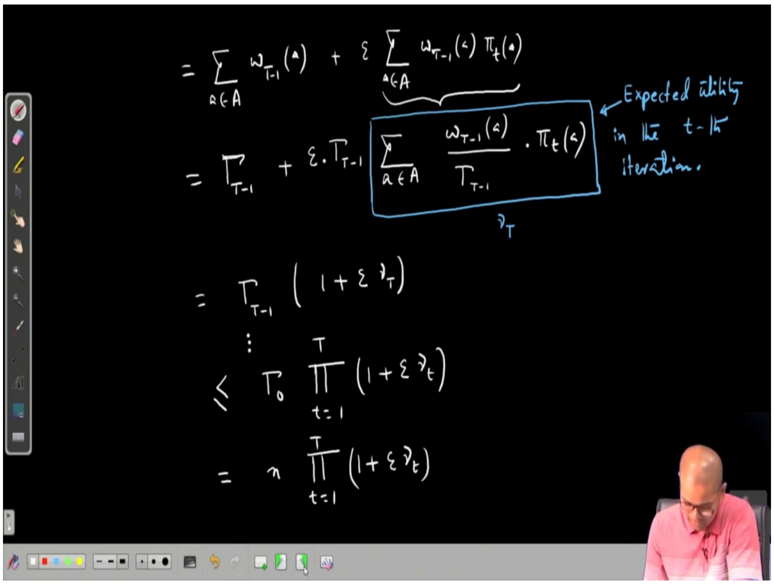


So, what we have let us write we had till now $\Gamma_T \geq \prod_{t=1}^{T}(1+\epsilon)^{\pi_t(a^*)}$. Now, I replace this product with sum in the exponent. So, this is same as $(1+\epsilon)^{\sum_{t=1}^{T}\pi_t(a^*)}$ and what is in the exponent, this is nothing but OPT. So, what we get is that $\Gamma_t \geq (1+\epsilon)^{OPT}$. So, I have related gamma with OPT that is equation 1.

Now, I will relate gamma with the expected utility. So, let us see, again let us start with $\Gamma_T = \sum_{a\in A} w_T(a)$ some of the weights, expand it. What is $w_T(a)$? It is $w_{T-1}(a)(1+\epsilon)^{\pi_t(a)}$. Now what I do is that this is less than equal to I use an upper bound for $(1+\epsilon)^{\pi_t(a)} \leq 1+\epsilon\,\pi_t(a)$ ok.

So, this I will check, I will let you verify that $(1+\epsilon)^x \leq 1+\epsilon\,x$ if $x\in[0,1]$. This I will let you verify, why less than equal to because I know I want to combine this equation 1 with whatever I get from equation 2 and the only way to combine is that I can get a lower bound for Γ_T. If I can fill up this thing Γ_T is less than equal to this something which is hopefully something related to expected utility then I can forget Γ_T and I can directly relate OPT with the expected utility of the player.

(Refer Slide Time: 17:29)

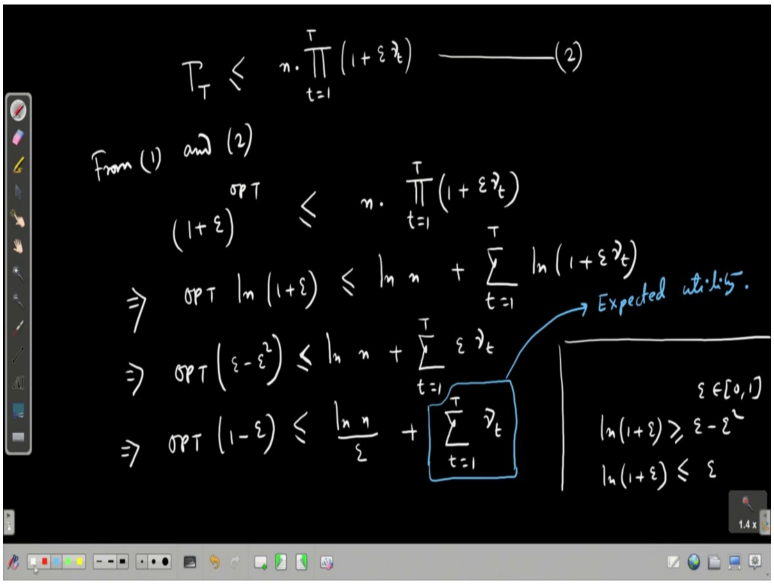


So, let us continue this. So, what do we have $a \in A$? So, $1+\epsilon\,\pi_t(a)$. So, this is let us unravel it ϵ comes outside $w_{T-1}(a)\,\pi_t(a)$, good this particular expression is very close to the expected utility let us see here. So, this inside part I just need the normalize normalizer $\Gamma_{T-1}(a)$. So, that I provide explicitly.

What is the first term? First term is Γ_{T-1} and for the second term I provide $\Gamma_{T-1}\pi_t(a)$. So, this particular object is exactly the expected utility of the player in the t-th iteration, this is expected utility in the tth iteration let us call it let us give it a name let us call it ν_t ok. ν_T it is a function of capital T. So, what we got is you can take Γ_{T-1} as common and you get $1+\epsilon\,\nu_T$.

And we can continue this process; that means, again I apply the same process to Γ_{T-1} and continue and at the end what I will get is that this is less than equal to this was less than equal to this less than equal to this was less than equal to $\Gamma_0 \prod_{t=1}^{T}(1+\epsilon\,\nu_t)$. And what is Γ_0? Γ_0 is the sum of the weights and all the weights are initialized to 1. So, this is $n\prod_{t=1}^{T}(1+\epsilon\,\nu_t)$.

(Refer Slide Time: 21:58)



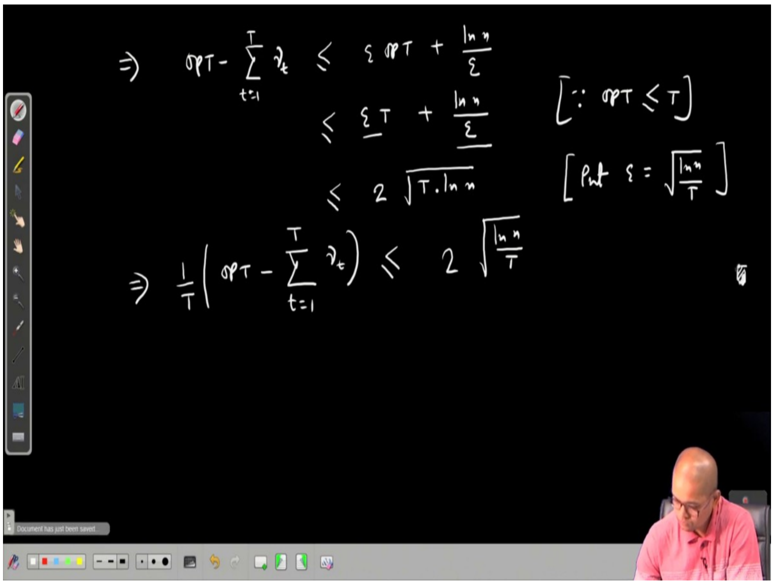
So, what do we get? We get Γ_T is less than equal to $n\prod_{t=1}^{T}(1+\epsilon\, v_t)$ this is our inequality 2. Now from 1 and 2, 2 says Γ_T is less than equal to something and 1 says Γ_T is greater than equal to something.

So, if I just ignore Γ_T I get that $(1+\epsilon)^{OPT}$ is less than equal to $(1+\epsilon)^{OPT}$ is less than equal to $n\prod_{t=1}^{T}(1+\epsilon\, v_t)$ ok. Very good, now we take let us take log on both side there are lots of products and exponents. So, $OPT\ln(1+\epsilon)\leq\ln n+\sum_{t=1}^{T}\ln(1+\epsilon\, v_t)$ ok.

Now, we will use. So, we will get rid of this logs by using standard inequalities this is $OPT\ln(1+\epsilon)$ for $\epsilon\in(0,1)$ $\ln(1+\epsilon)\geq\epsilon-\epsilon^2$ this holds for all $\epsilon\in(0,1)$ less than equal to $\ln n$ as it is t equal to 1 to capital T.

And $\ln(1+\epsilon)\leq\epsilon$. So, what I do is I apply the second one here. So, that I have this inequality going in the correct direction $\epsilon\, v_t$ ok. Now I want to separate out OPT. So, what I do is that I divide both side with epsilon then I get $OPT(1-\epsilon)\leq\frac{\ln n}{\epsilon}$ and without the epsilon the second term is $\sum_{t=1}^{T}v_t$ this is nothing but the expected utility in total.

(Refer Slide Time: 26:41)



So, I want to prove time averaged time averaged regret. So, that is OPT minus expected utility. So, what is OPT minus expected utility? $OPT - \sum_{t=1}^{T} v_t$ this is just rearranging terms $\epsilon OPT + \frac{\ln n}{\epsilon}$ ok. Now it is time to pick the epsilon and we will pick the epsilon which gives the this holds for all epsilon. So, we are free to choose epsilon which gives us the tightest bound. So, see and OPT is what, OPT is at most T this is $\epsilon T + \frac{\ln n}{\epsilon}$ since OPT is at most T, utility values are in between 0 and 1.

So, in every iteration the utility can be at most 1 and now you pick ϵ which minimizes the right-hand side that will give the tightest bond and this is a standard expression put $\epsilon = \sqrt{\frac{\ln n}{T}}$, put epsilon so that these two terms will be the same and then it is $2\sqrt{T \ln n}$. So, this is the regret and so the time average regret divide both side by T, $OPT - \sum_{t=1}^{T} v_t \leq 2\sqrt{\frac{\ln n}{T}}$ this is exactly what we need to prove ok.

Thank you. So, we will continue from here in the next class.