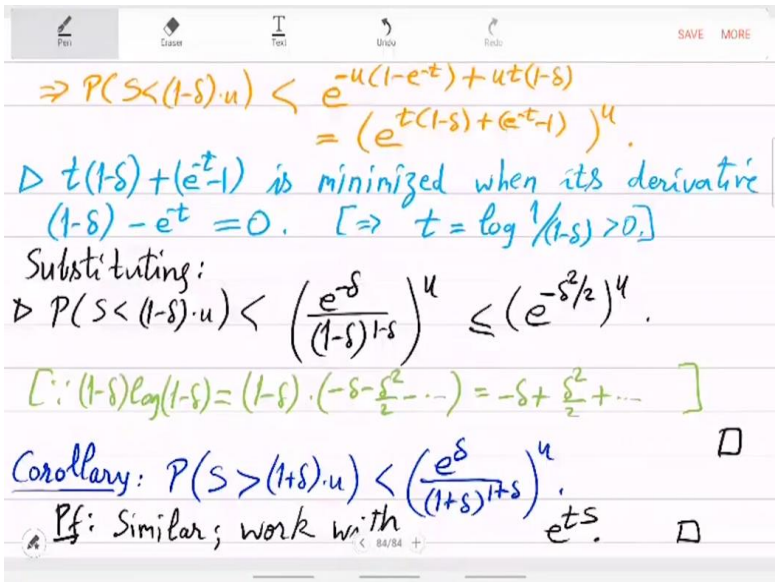
Probability for Computer Science
Prof. Nitin Saxena
Department of Computer Science and Engineering
Indian Institute of Technology - Kanpur

Module - 4
Lecture - 16
Chernoff's Bound. K-wise Independence.

Let us continue with this calculation. So, we want to now fix t; t is a positive unknown parameter.

(Refer Slide Time: 00:21)



How do you fix it? You fix it so that this right-hand side, it says it is an upper bound, right? So, you want to minimise it, which will tell you how small the probability is. So, let us do that. So, t 1 minus delta plus e raised to -t minus 1. This is minimised when its derivative which is 1 minus delta minus e raised to -t is 0. That is the simple calculation, which means that, in other words, your t is log of 1 over 1 minus delta; 1 over 1 minus delta is e raised to t.

So, for any delta, you will get this t, and it will be positive. That, you can see, because 1 over 1 minus delta is more than 1. So, this is positive. So, a positive t value exists; you use that; RHS is minimised; and to what? You get this. So, substituting, what you will get is probability that S is less than 1 minus delta u is less than; you will get basically e raised to minus delta divided by 1 minus delta 1 minus delta, whole thing raised to u.

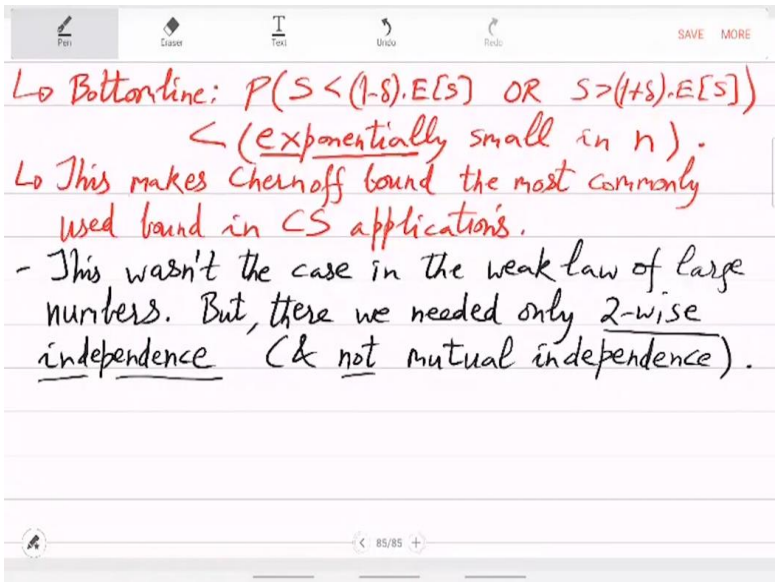
So, this inside part, this becomes e raised to minus delta divided by the t that we have fixed. So, this base is a function of delta only. You can also work with this expression, but there is a nicer expression; you can upper bound it by e raised to minus delta square by 2. Why is that? Basically, just look at 1 minus delta log of 1 minus delta expansion. So, that is, log of 1 minus delta is minus delta minus delta square by 2 minus delta cube by 3 and so on; which is minus delta plus delta square by 2 and positive things.

So, the difference between e raised to minus delta and e raised to this, is really e raised to minus delta square by 2, and then, more negative terms. So, we can use this kind of first order approximation. It is a nicer expression. And that gives you the statement of Chernoff bound. That finishes the proof. And moreover, you can get this corollary. So, what about S being bigger than expectation?

What if S is bigger than the expectation by a multiple of 1 plus delta? So, that, actually, you can do a similar calculation with the negative t. And you will get this expression e raised to delta divided by 1 plus delta to the 1 plus delta, whole thing raised to u. Proof is similar; just work with e raised to ts. So, in this proof, you started with e raised to -ts, in this original proof; for the corollary, you work with e raised to ts.

And e raised to ts will be greater than e raised to t 1 plus delta mu. And then, you can apply Markov; and then, you can get a similar expression; and then, you can also simplify it if you want; but, after all this calculation, what did you learn in the end?

(Refer Slide Time: 05:52)

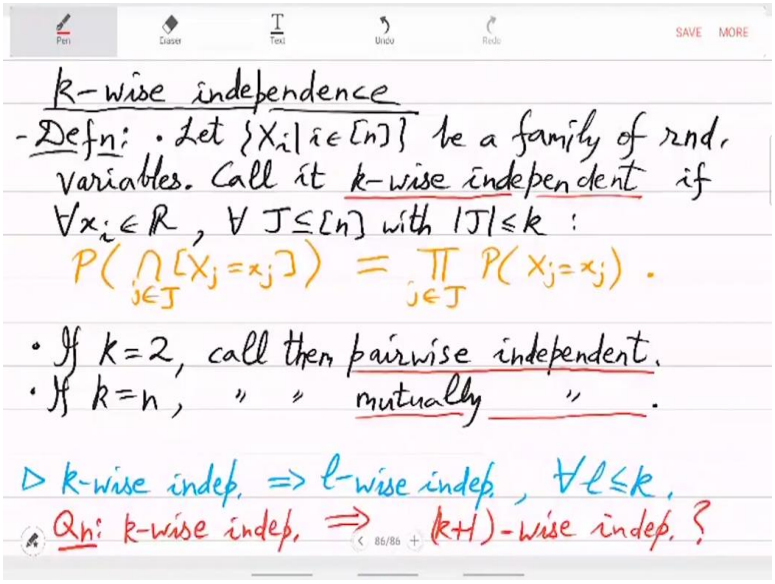


So, what you learn is this very important phenomena that probability, the sum of completely independent random variables distributed identically being very small or very large. Being very small or very large compared to the expectation is, this probability is less than some exponential in n. So, as you do more and more experiments, the chances are almost nil. So, your result will really be; result S or some S will be very close to expectation.

So, again, something that we have been repeating with every slide; but the point, keyword here is exponential. That is the new thing. So, this makes Chernoff bound the most commonly used bound in CS applications; be it theoretical or practical, Chernoff bound is used almost everywhere. One final remark; this was not the case with the weak law of large numbers, but there we needed only 2-wise independence and not mutual.

So, Chernoff bound is stronger because the assumption is also stronger. It is mutual independence; pairwise independence will not do. Now, to understand this distinction, we have to go through a bit of theory, so that you can really appreciate this distinction. So, let me do that next.

(Refer Slide Time: 09:32)



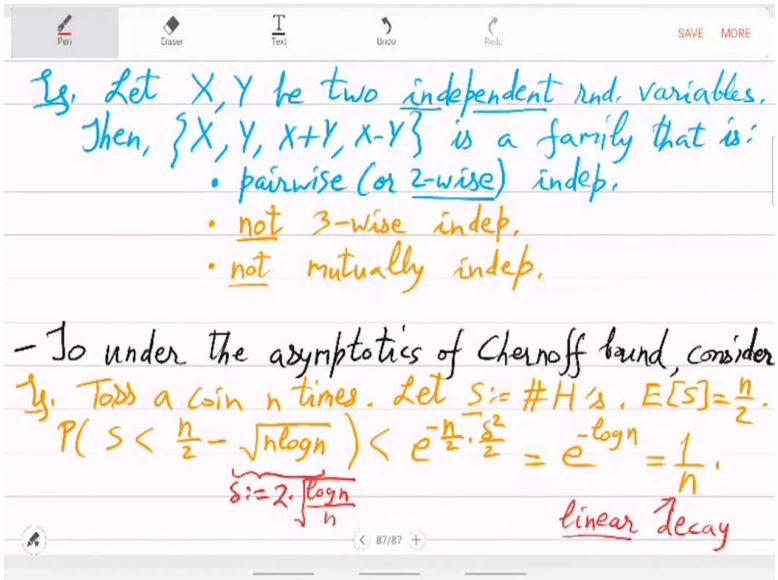
Let us do k-wise independence. This is also useful in computer science. So, what is k-wise independence? Let X i be a family of random variables; call it k-wise independent, if any k of them are, every k subset is independent. So, what do we mean by that? The following equation. So, if for all X i's real numbers and for all subsets of 1 to n that are small, at most k in size, if you look at the probability of intersection of these events.

So, big X j equal to small x j is the event that the random variable takes this value, following the probability mass function. And this happens for all the random variables in this subset big J. So, what is the probability of this? This cannot exceed the product probability, but in the case of k-wise independent, it will be maximised to the product. So, it is equal to product probability. That is it.

That is the most natural way to define k-wise independence, that any k subset that you choose of your random variables, the way they behave is totally independent. So, in particular, the intersection probability is just product. So, if k = 2, then call them pairwise independence. If k is equal to maximum, which is n, then call them mutually independent. That is the formal definition.

And you can immediately see that k-wise independence implies l-wise independent for all l less than equal to k. This is clear, because of this product definition. So, you can always go down, but can you go up? Can you go from k to k + 1? So, thus, k-wise independence imply k + 1 wise independence. That is not clear. Intuitively, it should not be true, because it seems to be requiring extra information than what you are given. So, let us see a proper example. And then you will understand why this discussion was important.

(Refer Slide Time: 14:24)



So, let X and Y be 2 independent random variables. Then, what can you say about the collection X, Y, X + Y, X - Y; collection of these four. What can you say about this family of random variables? Notice that X + Y and X - Y are obviously random variables. In fact, any

function is a random variable of X, Y. So, are they k-wise independent for various k. So, this is a family that is pairwise independent or 2-wise independent.

Why is that? Because, any two that you take, X or Y; X or X + Y; X or X - Y; Y, X + Y; Y, X - Y; or X + Y; X - Y; they are independent. You cannot deduce the second value from the first value, and that follows from the independence of X and Y themselves; from here it follows. However, this family is not 3-wise independent, because, if you take X, Y, then X + Y's value is implied. So, intuitively, just because of that, they are 3-wise dependent.

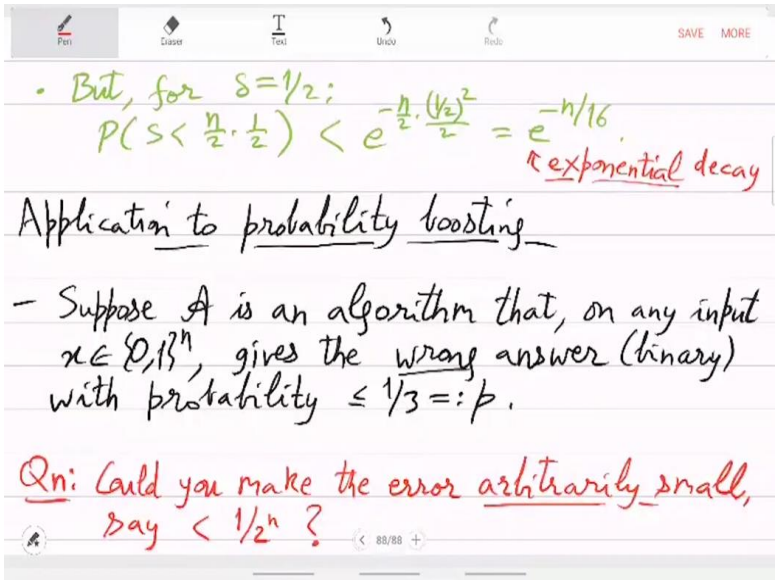
And hence, it is not mutually independent, because mutually would actually require 4-wise independence, but that certainly is not there; because, otherwise, 3-wise would have been there. That is a simple example; tells you that very easily this Y's independence can change. And there is a big gap between pairwise and mutual independence. So, that is the gap between law of large numbers and Chernoff bound, in the hypothesis of the two.

Let us go back to Chernoff bound. So, to understand asymptotics of Chernoff bound, consider the following example. So, toss a coin n times. Let S be the number of heads. So, what is expectation of S? Well, every toss is independent; these tosses, n tosses are mutually independent. And probability, each time is half of head coming; so, it is n by 2. It would have been n by 2 even if they were dependent; but anyways, we are independently tossing them.

So, expectation is n by 2. Now, let us look at the probability that S is smaller than n by 2 minus square root of n log n. So, can the number of heads be away from the mean with expectation n by 2 by a square root? So, what is delta here, if you want to apply Chernoff bound? So, delta is; you can see, it is log n by n square root. Basically, it is the ratio of the 2; this square root n log n by n by 2. That gives you delta.

So, delta is very small; as n grows, it is actually very small. It is a function of 1 over square root n. And hence, Chernoff bound gives you minus n by 2 delta square by 2, which is same as; you can see, it is e raised to minus log n, which is 1 by n. So, that is a linear decay. So, if you are talking about discrepancy being this square root n, then the decay is linear. It is not very fast, but it is also not bad. This is like the law of large numbers. So, that was point number 1.

(Refer Slide Time: 20:56)



Next is more interesting. But if you take delta to be half; so, half means that number of heads is n by 4. What about that? So, you expect n by 2, but suppose you got n by 4 heads? So, what was the probability of this event happening? So, this is by Chernoff bound, e raised to minus n by 2 and delta square by 2. So, you get e raised to minus n by 16, and that clearly is an exponential decay. And that is the strength of Chernoff bound.

This is not something which this law of large numbers can give you. So, this is the gap between linear and exponential. In this case, what is happening is, you are looking at an event which is really far away from the expectation. And then, you are applying Chernoff bound with mutual independence. So, you get actually very low probability as n increases. So, this is the difference between the two kinds of bounds.

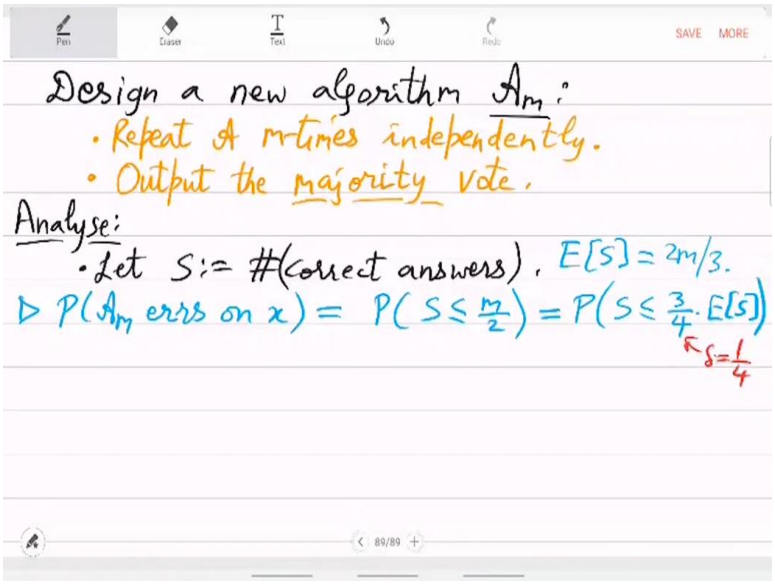
And this difference is actually very important in practice, and let us see that. It has an application to probability boosting. What is probability boosting? So, this happens in randomised algorithms. Suppose A is an algorithm solving your favourite problem, and say it is a very fast algorithm. It solves your favourite problem; the only issue is that it makes errors.

So, on input X, let us say of bit size n, gives the wrong answer, which is a binary answer. So, the algorithm will only say yes or no, but it will give the wrong answer with some probability, let us say one-third. That is the error probability p. So, now, one-third may seem like a large amount of error. You wanted an algorithm that never makes any mistake; one-third seems too large, a 33% error.

So, when you run the algorithm, you get an answer; you have little confidence. You want more confidence. You want, let us say, confidence more than 99%, which you are not getting here. So, could you get there? How do you get there? That is the question of boosting. So, could you make the error arbitrarily small? As small as you want; as small as you have; the more time you have, the smaller error you want to make.

So, say for concreteness smaller than 1 over 2 raised to n. Remember that already the input length is n, and you are asking for exponential in that; the error to be; from one-third you want it to be reduced to that much; which will be like 99.99999% of confidence in the answer. So, could this be done?

(Refer Slide Time: 26:07)



So, fortunately, Chernoff bound is there to help you. So, design a new algorithm; call it A m. What will it do? So, it will repeat A m times independently. So, m times means, you will get m bits, 0 or 1? And now, what do you think the output should be? So, the output naturally would be, a bit that appears the most. That is the majority votes; output the majority vote. So, it is like elections in a democratic setup.

Every person votes and, let us say, votes independently. And then, in the end, what is the majority? So, the bit which appeared just above half many times, 50% of the time, wins. And you say that, okay, that is the answer I am going with. Seems like a reasonable algorithm. Now, what is the advantage of this? Does it have any advantage? So, that would be in the analysis. Let us analyse this.

So, let S be the number of correct answers. Now, the probability that A m errs on input x, that will be when S; that is the number of correct answers is less than half. So, that is the event that S is less than equal to m by 2. So, these independent m iterations, more than m by 2 of them were wrong. That is the bad case. So, S is less than equal to m by 2, which is; now, what is the expectation of S?

Expectation of S is 2 m by 3, right? Because the wrong answer, it comes with probability one-third. So, correct answer with two-thirds; that is the expectation. So, compared to the expectation, what is this? This is three-fourth of the expectation. So, the delta that you are going with is one-fourth. So, what is the chance that S is below the expectation by a multiplicative factor of one-fourth? That is what you have to analyse using Chernoff bound. So, we will finish this next time.