Linear Programming and its Applications to Computer Science

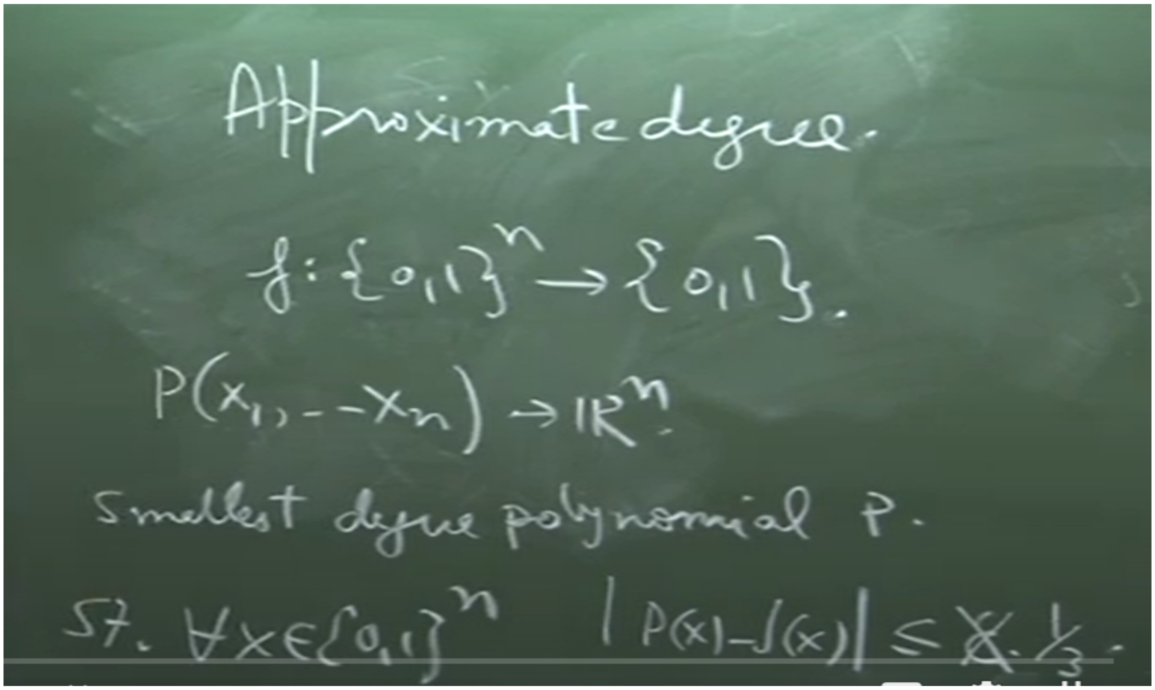
Prof. Rajat Mittal
Department of Computer Science and Engineering
Indian Institute Of Technology, Kanpur

Lecture – 09
Approximate Degree as LP

Let me give you more exercise and this is going to be slightly more difficult. This is an exercise for you. So ready with pen and paper. This is the problem of what we call approximate degree. Ok. What is approximate degree? Suppose you are given a Boolean function that means it takes an n bit string and outputs of it. Ok and now you want to write it as a polynomial.

What does it mean? You want to write it as a polynomial in x1 to xn such that the value here matches at all the 2 to the power n possible inputs. It is clear that there are 2 to the power n possible inputs that many strings my polynomial and you can do it but that is kind of expensive in the sense that you need a very high degree polynomial. So, what people are interested in is it possible to approximate this function with a polynomial which has small degree. The question is clear.

What does it mean? You want degree polynomial p such that for all x let us say the difference between px and fx is less than some constant and for are E's we will just say 1 by 3. Ok, so, this is first the notion of approximate degree. So, let us take some time to absorb it. Once more we are given a function we want to understand what the best in some sense polynomial representation is. Ok, actually lot of it is known that if you want to have a polynomial representation of it of this certain kind then there is a unique polynomial.



Approximate degree.

$f:\{0,1\}^n \to \{0,1\}.$

$P(x_1,--x_n) \to \mathbb{R}^n!$

Smallest degree polynomial P.

St. $\forall x\in\{0,1\}^n \quad |P(x)-f(x)| \le \cancel{\epsilon}. \frac{1}{3}.$

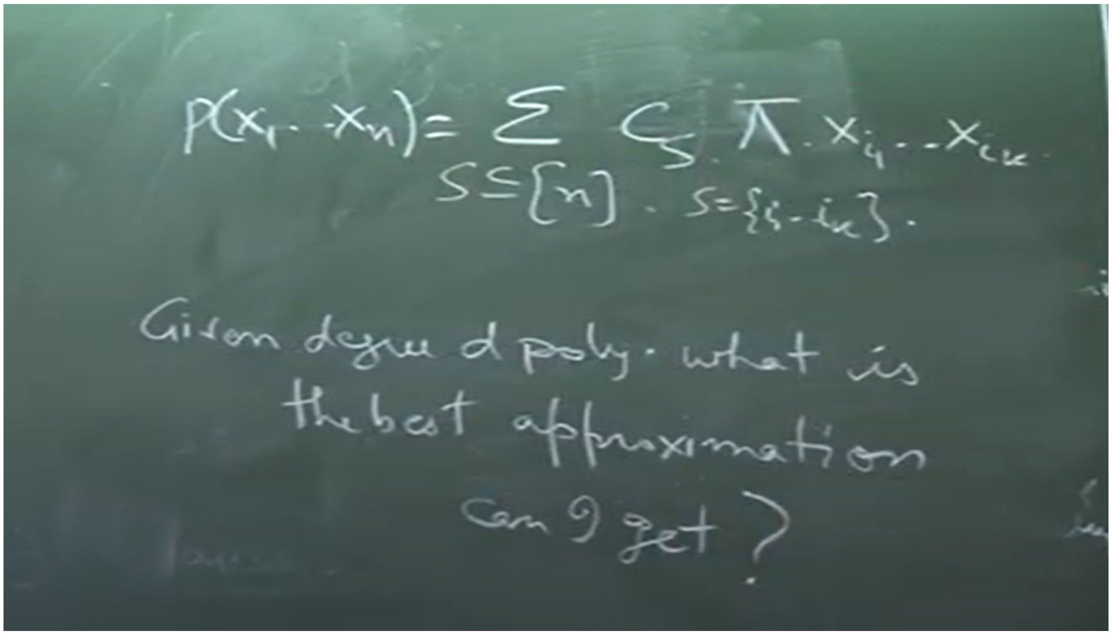
So, remember now your x1 to xn are real. So, this is a polynomial where your coefficients are coming from real. In spite of that there is a single polynomial which gives you which agrees with this function on all n points. This is one way in which you can say that your polynomial represents a function, but this is very expensive in the sense that you need a high degree. So, what people mathematically are interested in is, is it possible that there is a small degree polynomial which is close to f.

What does it mean close to us? f for every input it should not be more it should not be away from fx by more than 1 by 3 distance. Clear and now we are interested in finding a smallest degree polynomial. Now one thing to notice is since your input is only 0, 1 I never need a degree higher than 1. My x1 square is equal to x1. I just want to make sure that my polynomial agrees at these points.

So, I am only interested in the value at these points. So, there is no point having x1 square in my polynomial. I can just replace it by x1 and get the same thing on all of these to the power n points. Right So that means my polynomial if you think about it, it will look like some coefficient of s multiplied by the monomial where s. So there are at most 2 to the power n terms in my polynomial.

Is this point clear to you? This is just to emphasize what we are doing. Just to tell you this has problems in giving circuit lower bounds, secret sharing schemes, learning theory, quantum computing. There are surveys about this approximate degree and it has applications in many areas in theoretical computer science. So, now this is a problem and one of the reasons why this is interesting is because there exists a linear program for it.

Now can you make a linear program for it? What would you want to minimize? This is going to be an issue.



$$p(x_1 \cdot\cdot x_n) = \sum_{S \subseteq [n]} c_S \cdot \prod \cdot x_{i_1} \cdot\cdot x_{i_k} \quad S = \{i_1 \cdot i_k\}.$$

Given degree d poly. what is the best approximation can I get?

Now you realize that degree is integer. So minimizing degree you should kind of realize that oh that does not look like a linear thing that is going to be a problem. So we are going to slightly change our problem and we are going to say given all degree polynomials of this kind because you do not need higher degree. What is the best approximation can I get? And then I will say oh for 100 do I get 1 by 3, for 200 do I get 1 by 3 and as soon as I get 1 by 3 I will say I am done. Right, so in terms of polynomials this is not very expensive because I just have to repeat my procedure n times.

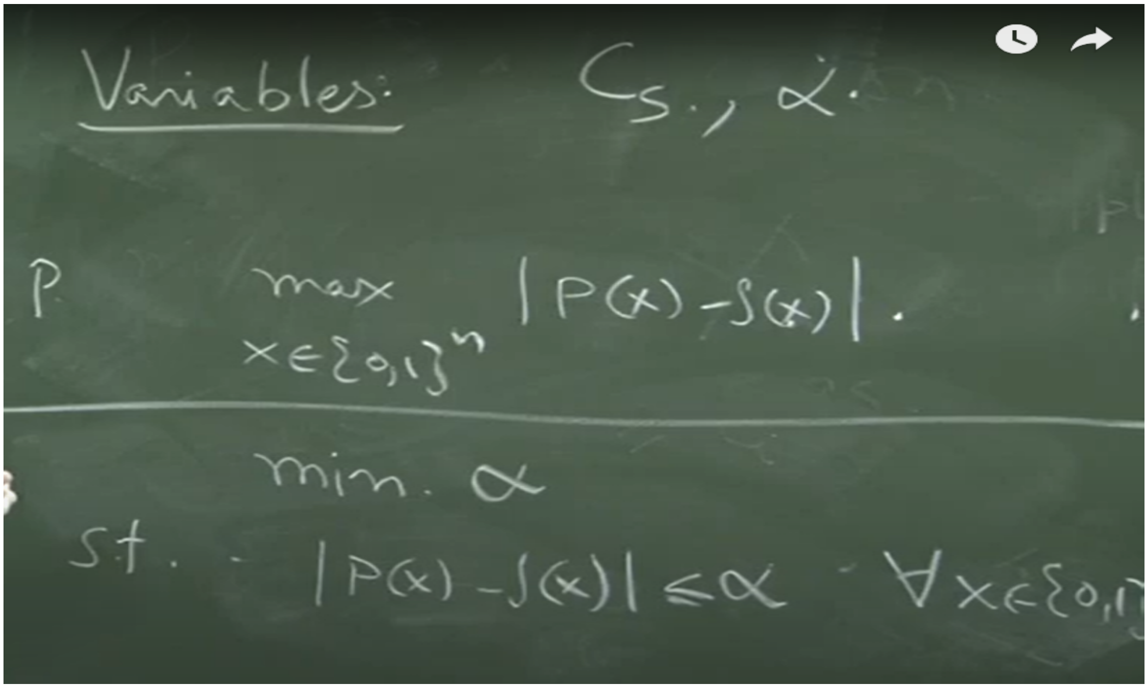
So this is ok, this is number of variables. This is still one extra n, so it is not very expensive. But now this can you write it as a linear program? My claim is you can. So write a linear program what are the variables? What are the constraints for this problem? And this is interesting. This is not that straightforward and I would be happy to explain any doubt here in terms of the definition of.

Remember this is a polynomial in r to the n that means the coefficients are coming from real numbers. Right. But we only need that it works fine on these Boolean inputs. So, I want to create a linear program whose answer will give me what is the best possible approximation like. So best possible approximation is like the maximum value of this for any degree-d polynomial. I am not worried about how big my linear program is, but is there a linear program which captures that problem? Right, so you can think of it as what is the error for a polynomial p? The error is this.

And I want to find out the best degree-d polynomial which, so this goes in kind of two steps. Generally when you define the approximate degree you first fix a constant. You say that I want to be this close to my function. 1 by 3 is an arbitrary choice. It could be 1 by 2, it could be 1 by 100 depending upon your application.

And it turns out that that does not affect the degree by much, but that is a separate proof. Now you have fixed 1 by 3, you want to find the smallest degree. That is kind of hard because you are optimizing degree which kind of goes in the integer way. I cannot put it in my constraints. I cannot put it in my optimization problem.

So I flip the problem. Instead of saying that I want this to hold true, I say let me look at all degree-d polynomials and find out what best approximation I can get. And then I will vary d and I will stop as soon as I get less than 1 by 3. So here your duality is English duality, not the mathematical linear program duality. Yes you should be very careful now, specially speaking to me when you use the word dual. Because



Variables: C_S, α.

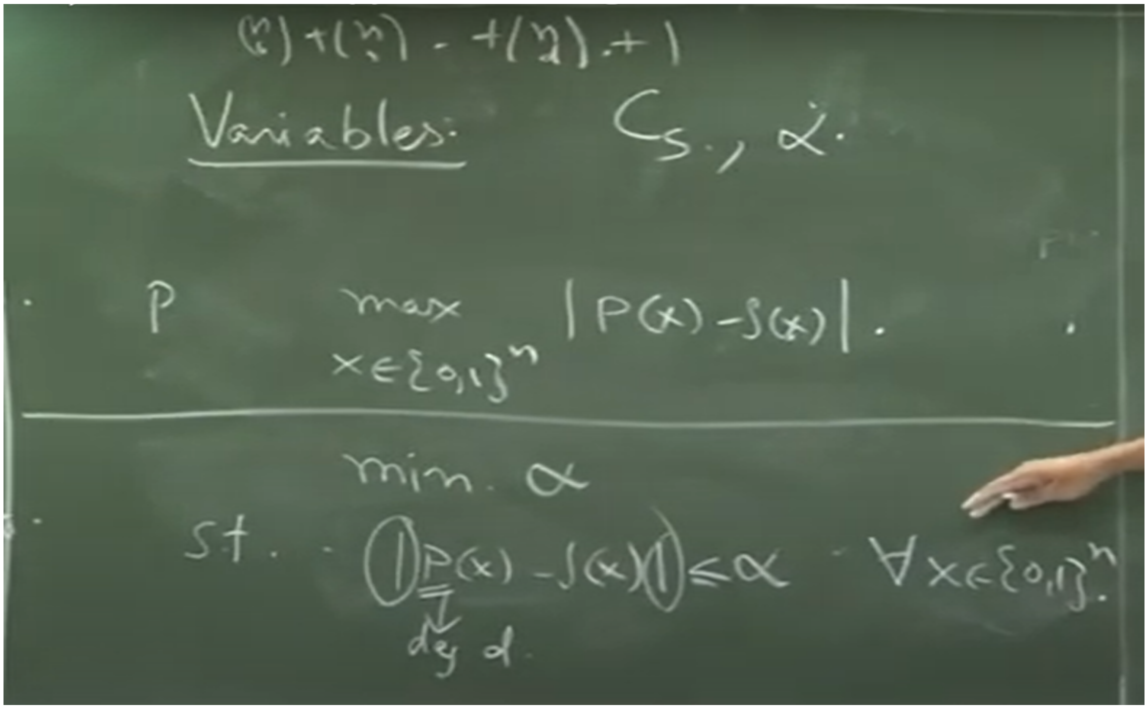
P $\max_{x \in \{0,1\}^n} |P(x) - f(x)|$.

min. α

s.t. $|P(x) - f(x)| \leq \alpha \quad \forall x \in \{0,1$

so what should be the variables? The Cs you mean. Have you seen the program before? No, ok. Approximation bound, how do you want to represent it? Alpha. So now if these are the variables, what is your linear program? I know you are close to it but let us see if someone else can answer. You want to minimize something.

Minimize bound means alpha. Right, yes. So you want to minimize alpha. Constraint is, is that what you were saying? Sorry. Yes we will go to that but It is true for all x n-bit strings. I is that all? No, we are not worried about 1 by 3, you just want to find the best possible approximation here. Is that all? Though, you are missing something here. No


$\binom{n}{0}+\binom{n}{1}-+\binom{n}{d}+1$
Variables: C_S, α.
P $\max_{x\in\{0,1\}^n} |P(x)-f(x)|$.
min. α
s.t. $|P(x)-f(x)|\leq\alpha$ $\forall x\in\{0,1\}^n$.
deg d.

That you can put if you want but you know linear program will take care of it. What else? Ok But I have not put the constraint that Ci is greater than equal to 0. No, no, no it would not. If it just was outputting 0 or 1 then the degree would be very high. The point is now it can even answer point 9 and it is ok. If Fx is 1 then I am fine with the polynomial outputting 0.9. But ok. So how many variables are there? What is T? D. No, right, so the point is the one thing which is not clear here is that this is only a degree D polynomial. So then remember even though it looks like this, x here is not a variable.

I have written it like this but x is not a variable. The variables are the coefficients and they will only go up to the D size subsets. So that would be like N to 0 plus N to D. So I have these many variables plus 1 for alpha. Is this a linear program? Not yet because of absolute values.

But I am sure all of you can convert it into two constraints both of which are linear. Ok. And I do not know about many other fields but at least for quantum computing this realization was very very important. Kind of looking at this, changing the problem to the dual nature and then writing out this thing. This is the only way in which we can bound approximate degree for non-symmetric functions. This is the only technique we know and people have been working on it for at least 20 years.

So at this point you are thinking of solution because you are thinking I will feed this linear program into a linear programming solver. But that is not what is expected here. What happens here is that you are given a function like AND or parity and you want to find out whether its approximate degree is root N or N or N to the power 2 by 3. So it is never that you are going to feed this into a linear programming solver. That will never give you the asymptotic bound.

It will only give it to you for one particular N and that does not give anything. So just this linear program you can write the dual of it equivalence and then you can make many things with it. So even though and this is another very nice example where you never actually solve the linear program but the mathematical structure, the beauty of it which I have talked about allows you to use this linear program to give lot of information about these things and we will see it.So generally our idea is oh, create a linear program put it to the solver.

That is not it. Ok Like a general stupid course linear program will just tell you simplex method, take the linear program put it into simplex method but there is much more to it and that is why we have to take the comprehensive view of what linear program is and we will talk about it. Did I answer your question? So yeah I am not worried that I cannot solve it by putting it in linear programming solver. I am just interested that it is a linear program.