Approximation Algorithm

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 07

Lecture 32

Lecture 32 : Randomized 1/2 Factor Approximation Algorithm for MAX-SAT and MAX-CUT

Welcome. So, in the last couple of lectures we have been seeing the idea of deterministic rounding to get approximation algorithms to design approximation algorithms. So, from this lecture we will start seeing another very interesting and useful idea which is randomized rounding of linear programs to get approximation algorithms to design a good small factor approximation algorithm. And again the idea is to see couple of examples to understand how this idea is implemented. So, this topic is randomized rounding, random sampling basically using randomness and randomized of linear programs to design approximation algorithm. So, our first example is the MAX-SAT problem and we will see a very easy approximation algorithm randomized approximation algorithm MAX-SAT.

given m clauses $C_1, \ldots, C_m$ over n variables where C_i is logical OR of some number of literals compute a Boolean assignment to this n variables to maximize the number of clauses satisfied ok. And the algorithm is very simple what we are going to do? We will set each variable to true or false with equal probability. set each variable to true or false with equal probability independent of everything else. for randomized algorithms in the worst case the performance could be bad.

So, what we look at is the expected performance. Let ALG be a random variable denoting the number of satisfied clauses. ok because it is not fixed the algorithm is random. So, ALG will take various values with various probabilities. We were we are expected in computing expected ALG expectation of ALG want to compute expectation of alg.

This is typical in randomized algorithms because there could be small very small probability with which the outcome is very bad, again there could be very small probability where the outcome is very good. So, it makes sense to look at the average sort of performance which can be formalized using the idea of expectation of the random

variable. So, for that we introduce random variables to compute expected expectation ALG. So, for all $j \in [m]$ we have M clauses let Z_j be the indicator random variable for the event that the clause C_j is satisfied ok. That means, what? That means, Z_j is 1 if C_j is satisfied and 0 otherwise. So, what we can see is ALG which is the number of clauses satisfied is $Z_1 + Z_2 + ... + Z_m$ ok.

So, we want to compute expectation of ALG. which is expectation of $Z_1 + Z_2 + ... + Z_m$ observe that clauses can share variables. So, this random variable Z_1, Z_2 and so on they are not independent. but the linearity of expectation property or that lemma does not get independence. So, even if they are they may be dependent in any way linearity of expectation says that expectation of sum of random variables is sum of expectation of individual random variables.

That means, I can push this expectation inside the sum. that is this is $E[Z_1] + E[Z_2] + ... + E[Z_m]$. So, all I need to do is to compute this individual expectations. So, for that let us investigate any particular random variable Z_j. So, let C_j the j-th clause contains l_j literals.

So, what is the probability that C_j is satisfied by the random assignment. You see for every literal there are l_j many literals and if any one of them is said to true then it is satisfied. That means, C_j is satisfied is any of these l_j literals is said to true. that means, the only way that the C_j is not satisfied is that all the l_j literals is set to false. For example, suppose if you look at a clause say $x_{10} \vee \bar{x}_{12} \vee x_{20}$ the only way that this clause is not satisfied is to set each literal to false that means, setting x_{10} to false x_{12} to true and x_{20} to false.

If we are setting each variable to true and false with equal probability what is the probability that all the literals are set to false this is $1 - (\frac{1}{2})^{l_j}$. Now, l_j is at least 1. So, this is greater than equal to half. Now, what is expectation of Z_j Because Z_j is a indicator random variable for indicator random variables expectation is same as the probability of the event that you can remember and also easy to prove. So, let us first see.

So, Z_j takes value 1 and if $C_j$is satisfied plus it takes value 0 if it is not satisfied ok. So, this is greater than equal to half. So, what we have seen is expectation of Z_j is greater than equal to half for all j this is for all $j \in [m]$. So, then what is expectation of ALG? Expectation of ALG which is sum of expectation of Z_js. this is greater than equal to m by 2. Now, $m \geq opt$ because at max I can satisfy all clauses.

So, this is greater than equal to $\frac{opt}{2}$ since $m \geq opt$. So, our algorithm has an approximation factor of 2. why we write at most 2 and not 2? What we have basically shown is it is at most 2 maybe there is some analysis which can show that it is better than 2. So, that is why we write our approximation factor of our algorithm is at most 2. Now, interestingly this same analysis can be extended to weighted version. So, what is weighted version of this MAX-SAT in weighted max at each clause has a positive weight.

the goal is to compute an assignment which maximizes sum of weights of the clause satisfied ok. So, you take it as a homework. to show the exact same algorithm that is setting each variable to true or false with equal probability independent of everything else is a half factor approximation algorithm is a randomized half factor approximation algorithm. setting each variable to true or false with equal probability achieves an approximation factor of at least half. for weighted max set ok.

So, the same algorithm if we analyze for MAX3SAT then we see that for MAX3SAT if l_j is 3 then this is equal to $\frac{7}{8}$. So, this is this will be equal to $\frac{7}{8}$ if l_j equal to 3 and it turns out this is the best approximation factor achievable for MAX3SAT under standard complexity theoretic assumptions. So, let us write our algorithm achieves an approximation factor of $\frac{7}{8}$ for weighted MAX3SAT that is each clause is a logical OR of exactly 3 literals. and it turns out that this is the best known under P not equal to NP hypothesis and this is the landmark result in complexity theory theorem. The proof is beyond the scope of this course.

If there is $\frac{7}{8}+\epsilon$ factor approximation algorithm even for unweighted MAX3SAT for any constant $\epsilon > 0$. $P \neq NP$. So, if you assume that $P \neq NP$ then the best approximation factor is basically $\frac{7}{8}$. So, you can have an approximation algorithm of say $\frac{7}{8}$ plus some factor which is $o_n(1)$ maybe say $\frac{1}{\log n}$ that is not pulled out, but effectively for large enough instances it is $\frac{7}{8}$. So, our next problem is weighted max cut.

Here what is the problem? Given age weighted graph, age weighted undirected graph $G=(V,E)$ and weight function from E to R weights we will assume non negative compute a subset s of vertices such that the cut edges sum of weights of cut edges. So,

$\sum_{e \in \delta(S)} w_e$ is maximized. Recall water cut edges $\delta(S)$ is the set of edges whose exactly one end point belongs to S, they are also called boundary edges. For this problem also it seems that there is a very natural easy randomized algorithm start with S equal to empty set, put every vertex $v \in V$ in S with probability half again independent of everything else. following similar kind of analysis it can be shown that it is the half factor approximation algorithm again I am giving it as a homework.

Show that the approximation factor of our algorithm is at least half ok, did I write at most somewhere yeah this is at this should be at least half. for maximization problems approximation factors are less than 1 and the higher the approximation factor it is better ok. So, let us stop here. Thank you.