Approximation Algorithm

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 07

Lecture 34

Lecture 34 : Flipping Biased Coin for Better Than .5 Approximation Algorithm for Max-SAT

So, in the last class we have seen the method of conditional expectation for de randomizing randomized algorithms and this technique is quite general it can be used for de randomizing various randomized algorithms many of them discussed in the in this course and we will mention it and you can take it as a homework to use the method of conditional expectation to derandomize them. we will mention when it is the case when it is easy to randomize it. So, again we resume our journey to see randomized sampling for designing approximation algorithms and again we revisit the MAX-SAT problem. So, can we look at the MAX-SAT problem. So, let us study the weighted MAX-SAT.

And, what is the randomized algorithm? We simply set each variable to true or false with equal probability. So, one can ask is it the best thing? or setting the variables to true with probability p for some value of p in between 0 and 1 and false with probability $1-p$ can give us better bound. So, we will see that we can choose p more judiciously to get better approximation guarantee. So, the idea is you flip biased coins.

So, for simplicity let us assume that there is no clause which is which contains exactly one literal and that literal is a negated literal. That means, there is no clause which is just say $\bar{x}_{10}$. So, we will remove this assumption later. So, assume not without loss of generality. will see later how we can get rid of this assumption.

Assume there is no clause which is ah negation of one variable ok. So, single variable clause are allowed, but they should appear positively. So, what is the algorithm? Again we set each variable x_i to true with probability p and false with probability $1-p$ independent of everything else ok. So, let us see what we have achieved. So, we can prove this lemma.

our algorithm satisfies each clause with probability at least minimum of p and $1-p$ square assuming that assuming that there is no clause which is just a negation of one variable ok. let C_j be any arbitrary clause. There can be 2 cases that there exist a variable which appears positively in C_j case 1. there is a variable that appears positively in C_j. that variable is set to true with probability p our algorithm sets that variable to true probability p.

Hence, C_j is satisfied with probability at least p which is in term in term greater than minimum of p and $1-p^2$. So, case 1 is resolved case 2. C_j does not have any variable appearing positively. Now, from our assumption if C_j does not have any variable appearing positively, then it cannot contain only one variable. So, by our assumption since C_j does not contain any positive variable, it must contain at least 2 variables.

appearing negatively. Hence, C_j is satisfied with probability at least again let us argue what is the probability that probability that it is not satisfied for it to be not satisfied all its variables because they are appearing negatively all of them should be said to false all of them are appearing negatively. So, all of them should be said to true. So, for C_j to be not satisfied all of its variables should be set to true and it has at least 2 variables. So, the probability that all of the variables is set to true is at most p^2 and so, hence the probability that C_j is satisfied is at least $1-p^2$ which in turn is less than equal to minimum of p and $1-p^2$.

So, this concludes the proof. So, the idea is we pick p. So, again the same analysis that our algorithm has a minimum of p comma $1-p^2$ approximation factor. the approximation factor of our algorithm is at least minimum of p and $1-p^2$ ok. Proof is again very similar what is ALG? ALG is recall we had indicator random variables.

and these are the weights $w_1 Z_1 + w_2 Z_2 + \ldots + w_m Z_m$. So, what is expectation of ALG? expectation of $w_1 Z_1 + w_2 Z_2 + \ldots + w_m Z_m$. Again we apply linearity of expectation this constant terms come out $w_1 E[Z_1] + w_2 E[Z_2] + \ldots w_m E[Z_m]$.

This is equal to w_1 times probability that expectation of Z_1, Z_1 is a indicator random variable expectation is the probability that C_1 is satisfied plus up to w_m times probability that C_m is satisfied. Each of this probability is less than equal to minimum of p and $1-p^2$. So, this is no this is greater than sorry at least each of this probability is at least minimum of p and $1-p^2$. Each clause is satisfied with probability at least minimum of p and $1-p^2$. This is greater than equal to minimum of p and $1-p^2$ $(w_1 + \ldots + w_m)$.

$(w_1 + \ldots + w_m)$ this is the upper bound on opt. So, this is greater than equal to minimum of p and $1-p^2$ times opt ok. So, we want to pick p so that minimum of p and $1-p^2$ is maximized. So, choose p so that minimum of p and $1-p^2$ is maximized and what we should choose? We should choose p so that $p=1-p^2$ because larger p increases p and smaller p increases $1-p^2$. So, the right balance will be the p which maximizes this minimum is the p where p and $1-p^2$ meet.

So, p equal to $p=1-p^2$. $p^2+p-1=0$. So, $p=\frac{-1+\sqrt{5}}{2}$. which is roughly 0.618. So, we have a 0.618 factor approximation algorithm because the approximation ratio is minimum of p and $1-p^2$. So, for p equal to $\frac{\sqrt{5}-1}{2}$ the approximation ratio of our algorithm is this p which is $\frac{\sqrt{5}-1}{2}$. which is roughly 0.618 which is greater than half. So, under that assumption that there is no clause which is just one variable appearing negatively we have a better than half factor approximation algorithm.

Next what we do we want to get rid of that assumption. So, for that we need a better bound on opt here we have used that opt is greater than equal to opt is less than equal to $w_1+w_2+...+w_m$. So, we need to improve this bound. So, how? So, for that again we make an assumption, but that assumption is without loss of generality. without loss of generality that for every variable index i the weight of unit clause x_i.

Unit clause means a clause which has just one variable which is x_i, if there is no such variable then the weight of that clause that weight we are saying 0. The weight of unit clause x_i is at least the weight of unit clause $\bar{x}_i$ ok. So, why this is without loss of generality? Because if it is not the case, if there exist an i such that the weight of x_i is strictly less than the weight of $\bar{x}_i$ then we can replace x_i with $\bar{x}_i'$ in all the variables and solve it and then you see we have this assumptions are made. So, this assumption is without loss of generality. Now, let v_i be the weight of the unit clause $\bar{x}_i$.

Now, whatever you set if x_i you set to true then you does not satisfy this unit clause $\bar{x}_i$ and hence you lose score or point of v_i from total $\sum w_i$ and other on the other hand if you set x_i to false then you satisfy this clause $\bar{x}_i$, but you does not satisfy the unit clause x_i and there you lose at least v_i because of this assumption which is without loss of generality. So, hence what we can see is opt is greater than equal to $\sum_{i=1}^{m} w_i$ this was the upper bound that sorry opt is less than equal to this was the upper bound i we used and for each variable it has to lose at least v_i. So, make it j because these are clauses. just for a cement respecting the semantics that we are using nothing mathematically wrong minus $\sum_{i=1}^{n} v_i$ ok.

Now, here is a. So, let me write the thing why this observation that for each i any assignment to x_i can satisfy exactly one of x_i. and one of unit clauses x_i and $\bar{x}_i$. Hence or the weight of $\bar{x}_i$ is v_i and the weight of unit clause x_i is at least v_i. So, any assignment the total sum of weights of the clauses satisfied is at most this. less than equal to $\sum_{j=1}^{m} w_j - \sum_{i=1}^{n} v_i$. So, now, we prove that this is again $\frac{\sqrt{5}-1}{2}$ factor approximation algorithm by choosing p equal to $\frac{\sqrt{5}-1}{2}$.

Setting p equal to $\frac{\sqrt{5}-1}{2}$ our algorithm achieves an approximation factor of at least $\frac{\sqrt{5}-1}{2}$ ok. And the analysis is exactly as usual like before with slight tweak. So, let us see ALG is again same as $w_1 Z_1 + \ldots + w_m Z_m$. So, expectation of ALG is expectation of $w_1 Z_1 + \ldots + w_m Z_m$. This is again I will apply linearity of expectation $w_1 E[Z_1] + \ldots + w_m E[Z_m]$ ok.

And expectation of Z_i as usual there each is greater than equal to minimum of p and $1-p^2$, but if I set p like this then minimum of p and p and $1-p^2$ is p. So, this is greater than equal to p times $w_1 + \ldots + w_m$ ah . So, here we need to use the fact that these holds only if these we cannot use these holds only if there are no unit clause with negated variables. So, for that what we let u subset of m be the set of clauses, set of all clauses except unit negated variables. So, what we do here we first write it as this is greater than equal to I sum over all those clauses in U $\sum_{j \in U} w_j E[Z_j]$.

Now, it holds because U this set of clauses does not contain any negative variables. So, this is greater than equal to p times $\sum_{j \in U} w_j$ ok. And this is what is exactly p times $\sum_{j \in U} w_j$ minus v_i because for each variable we are we are removing we are the unit clause which is the negation of the variable and the rest is u i equal to 1 to n, but this is greater than equal to opt this is greater than equal to p times opt this holds for p equal to $\frac{\sqrt{5}-1}{2}$. Recall this was minimum of p and 1 minus p square and for p equal to $\frac{\sqrt{5}-1}{2}$ p and $1-p^2$ is same. So, this is the p factor approximation algorithm without any assumption on the clauses ok. So, let us stop it.