Approximation Algorithm

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 12

Lecture 56

Lecture 56 : Introduction to Semidefinite Program

Welcome from last couple of weeks we have been seeing linear programming and its use for designing approximation algorithm. Today we will see another powerful tool which is called non-linear programming in particular a kind of non-linear programming known as semi definite program which is substantially powerful than the linear programs and we will see how semi definite programs can be used for designing approximation algorithms for polynomial time approximation algorithms. And again it boils down to the fact that we can solve semi definite programs in polynomial times like linear programs, but it gives substantially more power ok. So, let us start. So, semi definite programs . So, semi definite programming.

is a non-linear programming which is substantially more powerful than linear programming, but still can be solved in polynomial time. So, we will use semi definite programming to see better than half factor approximation algorithm for the max cut problem. Indeed this algorithm is the best known approximation algorithm for max cut problem. So, semi definite programming involves what are called positive semi definite matrices.

So, before seeing semi definite program let me explain let us see what are positive semi definite matrix. Definition a matrix $X \in R^{n \times n}$ is called positive semi definite. if for all vector $y \in R^n$ $y^{tr} x y$ which is a scalar quantity is greater than equal to 0. So, we will assume vectors to be column matrices. So, $y^{tr} x y$ is a scalar quantity which is greater than equal to 0 for all vector $y \in R^n$.

So, such a matrix is called positive semi definite matrix and you see this condition is bit difficult to verify. How can we check whether a given matrix X is positive semi definite because this condition requires going over all real vectors of dimension n. So, if X is a symmetric matrix then there are nice equivalent conditions which are easier to check. So, let us see the equivalent condition. So, here is a fact which we will use without proof these things can be a proved using linear algebra and I give it as a homework.

So, if $X \in R^{n\times n}$ is a symmetric matrix, then the following are equivalent. First one is X is positive semi definite. Second one is all eigenvalues of X are non-negative. third X is X can be expressed as $v^{tr} v$ for some $v \in R^{m\times n}$ for sum $m \le n$. Fourth condition is X can be expressed as $\sum_{i=1}^{n} \lambda_i w_i w_i^{tr}$ for some $\lambda_i \ge 0$ and orthogonal vectors orthonormal vectors $w_1, \ldots, w_n \in R^n$. What does orthonormal means? Orthonormal means that $w_i^{tr} w_i$ which is the inner product or the norm of w_i or the inner product of w_i with itself is 1. that is the normal means and ortho means this is for all $i \in [n]$. for w_i and w_j are orthogonal to each other. That means, $w_i \cdot w_j$ or the same as $w_i^{tr} w_j = 0$ for all $1 \le i < j \le n$.

ok. So, with this definition of semi definite matrices or positive semi definite matrices let us see what is semi definite programming. A semi definite program SDP in short is similar to a linear program that is the objective function and constraints are linear. In addition we can have an extra kind of constraint which are in terms of semi definite matrices. In addition we can have a constraint asking a square symmetric matrix. of variables to be positive semi definite.

So, let us see on canonical example of a semi definite program canonical example of semi definite program. So, suppose I again variables we will have a constraint which will say that symmetric matrix of variables is semi definite. So, often variables are entries of a matrix. So, variables are say x_{ij} $1 \le i, j \le n$. So, the constraint typically says as usual maximize or minimize linear function of the variables.

So, variables we denote by lowercase x subject to linear constraints of this variables $\sum a_{ijk} x_{ij} = b_k$ for all k. So, you have some number of linear constraints and we need symmetry these variables must be symmetric. So, $x_{ij} = x_{ji}$ for all i,j in between 1 and n and the matrix $(x_{ij})_{1 \le i, j \le n}$. must be positive semi definite which is denoted as $x_{ij} \ge 0$. So, this is the only constraint which is not allowed in LP.

but allowed in SDP ok. So, it turns out that modulo some technical condition if some minor technical conditions are satisfied semi definite programs can essentially be approximately solved. So, here is the fact. given minor technical conditions, the optimal value of an SDP can be computed within an additive error of ϵ in time polynomial in input size and $\frac{1}{\epsilon}$. which usually is enough for our purposes.

So, with slight lack of formality loss of formality we will assume that the optimal solution of an SDP can be computed in polynomial time. So, with slight loss of accuracy, we will assume that the optimal solution of an SDP can be computed in polynomial time. So, there is another equivalent formulation or equivalent perspective of semi-refinite programming equivalent way to look at semi-refinite programming which are called vector programming which is often useful. So, let us see what are vector programming. So, the variables here are the vectors here the variables are vectors.

$v_i \in R^n$ in n dimensional space they are the vectors and the objective function and constraints. are linear functions of the inner product these vectors. So, again a canonical example let us see canonical example of a vector program maximize or minimize a linear function of the inner product of the vectors. We have say n vector variables $c_{ij} = \langle v_i, v_j \rangle$ subject to linear constraint of the inner product of these vectors. So, we have say some number of constraints.

$a_{ijk} v_i \cdot v_j = b_k$ for all k, where these vectors $v_i \in R^n$ for all $i \in [n]$. So, vector programs are equivalent to SDPs, let us see how. So, suppose we have a SDP let us see what are these what are the equivalent vector program. So, let X be the symmetric matrix of variables of an SDP. So, X is positive semi definite by the equivalent condition if and only if $X = V^{tr} V$ for some matrix V.

So, given a solution X of the positive of the SDP, we can compute $V^{tr} V$ and this gives a solution to the i-th column of this V is the vector in the vector program. The i-th column of V be v_i. So, x_{ij} is nothing, but $v_i \cdot v_j$. So, you check this as a So, once if we have a semi definite program. So, what is the corresponding? So, if you have a semi definite program, what are the corresponding vector program? You replace x_{ij}'s with $v_i \cdot v_j$ everywhere and that is it and you forget this constraints.

So, you get from SDP and equivalent vector program or the other way if you have a vector program you replace each $v_i \cdot v_j$ with x_{ij} and add a constraint that $x_{ij} = x_{ji}$ and X is positive semi definite. So, with these two you can go back and forth from vector program to semi definite program and their solutions are same from one solution of from a optimal solution of vector program you can go get an optimal solution for SDP and vice versa. So, we will see that this vector program perspective of SDP is often useful while encoding or encoding a combinatorial problem as SDP and getting an approximation algorithm. So, let us stop here. Thank you.