CS294 A Toolkit for Algorithms **Spring 2010**

Lecture 1: January 20

Lecturer: Prof. Satish Rao *Scribes: Lorenzo Orecchia*

1.1 Problems and Open Questions

1. Explore applications of the recent "Quantum Algorithm for Linear Systems of Equations" by Harrow, Hassidim and Lloyd [5]. This algorithm can solve a linear system of the form $Ax = b$, in time logarithmic in the dimensions of A, an exponential improvement over classical algorithms. However, the quantum algorithm cannot output the vector solution x explicitly, but only give an approximation to the value of $x^T Mx$ for an input operator M.

2. The Arora-Rao-Vazirani algorithm (ARV) [1] is an approximation algorithm achieving a $\sqrt{\log n}$-approximation for the Sparsest Cutproblem. At the same time, there exists a simpler spectral algorithm, yielding a cut of value at most $O(\sqrt{\mu})$, where μ is the spectral gap of the instance graph. This can be thought of as an approximation ratio of $\frac{1}{\sqrt{\mu}}$. We are wondering whether it is possible to obtain a different approximation ratio for the ARV, i.e. one as a function of the spectral gap μ of the instance graph, for instance $\log(\frac{1}{\mu})$. Jonah claims this may be ruled out by the gap instance in [3].

3. Explore applications of the Multiplicative Weight Update method to problems in quantum computation. Recently, this yielded the result $QIP = PSPACE$ [6]. Is it possible to say anything about scenarios with multiple provers? Another direction is the exploration of a possible connection between the update methods and quantum evolution, and the Adiabatic Theorem.

4. The ARV algorithm has constituted a breakthrough in designing graph partitioning algorithm and a large effort has gone into finding fast versions of this algorithm. Currently, the best version only needs a polylogarithmic number of s-t maxflow calls and runs in time $\tilde{O}(n^{3/2})$. We would like to explore the question of whether this bound can be improved, in particular in light of the new techniques introduced by Spielman and Teng. A couple of directions are:

   - can the running time of s-t maxflow computations be improved?
   - more generally can we improve the number of iterations required by the interior point method for linear programming?
   - along a different direction, can we use electrical flows to understand the ARV better?

5. The Unique Games Conjecture (UGC). What approximations can we get within the Lasserre Hierarchy?

6. Lattice Cryptography

7. Quantum Set Cover

8. Quantum Money

1.2 Introduction

In these first lectures, we will give an introduction to semidefinite programs (SDPs), a class of optimization problems that will play a central role in the rest of the course. We start today by briefly discussing linear programs (LPs), defining SDPs and starting our discussion of SDP duality.

1.3 Preliminaries and Linear Programming

A crucial lemma to which we will appeal in our discussion of duality is the following Hyperplane Separation Lemma.

Lemma 1.1 *Given two convex subsets $A, B \subset \mathrm{R}^n$ having disjoint interiors, there exists an affine hyperplane $H_{h,c} = \{x : h^T x = c\}$ separatingA and B, i.e. such that $\forall x \in A, \forall y \in B$,*

$$h^T x \geq c \geq h^t y.$$

The main objects in linear programming are essentially systems of linear inequalities $a_i^T x \leq b_i$, where $x \in R^n$ and $a_i, b_i \in R.$ for $i \in [m]$. At the root of linear progamming duality lies the following theorem of alternative, which provides a characterization of when such a system is feasible, i.e. has a solution.

Lemma 1.2 (Farkas Lemma) *A system of linear inequalities $a_i^T x \leq b_i$, for $i \in [m]$, has no solution if and only if there exist $\lambda_1, \lambda_2, \cdots, \lambda_m$ such that $\sum_i \lambda_i a_i = 0$ and $\lambda_i b_i < 0$.*

This means that if the linear program has no solution, we can find a positive combination of the inequalities which yields a contradiction. In geometric term, this can be seen as a hyperplane separating the subspace $y : y = Ax$ from the point b.

A general LP is usually given as an optimization problem:

$$\begin{aligned} \max \ & c^T x, \\ \text{s.t} \ & Ax \leq b. \end{aligned}$$

The dual of this linear program is:

$$\begin{aligned} \min \ & b^T y, \\ \text{s.t} \ & A^T y = c, \\ & y \geq 0. \end{aligned}$$

These programs are related by this fundamental theorem.

Theorem 1.3 (Duality Theorem) *If either one of the primal and dual programs has an optimum solution then so does the other and the two optimum values are equal. The primal program is infeasible if and only if the dual is unbounded. The dual program is infeasible if and only if the primal is unbounded.*

Sketch of Proof Consider the case when both primal and dual are feasible. This theorem is shown in two parts. The first, known as Weak Duality, establishes that the primal objective value is at most the dual objective value. This is easy to see, given a primal-feasible x and a dual-feasible y as

$$c^T x \leq y^T A x \leq b^T y.$$

The other direction, known as Strong Duality, is harder because it actually entails producing a dual solution, indeed an optimal one, and is almost algorithmic at heart. In brief, one considers the feasibility problem given by the question: is the primal optimum larger than some value t? For t equal or greater than the primal optimum, this system is infeasible and we can apply Farkas Lemma to obtain a certificate of infeasibility. This certificate can be easily turned into a dual solution of value t.

References. For more about linear programs, see the Survey by Lovasz on the course webpage or check [4] and [2] online.

1.4 Semidefinite Programming

1.4.1 Positive Semidefinite Matrices

Definition 1.4 *For a symmetric matrix $A \in R^{n \times n}$ the foillowing are equivalent:*

1. *A is positive semidefinite, denoted $A \succeq 0$.*
2. *all of A's eigenvalues are nonnegative,*
3. *for every $x \in R^n$, $x^T A x \geq 0$,*
4. *A can be written as the inner product of matrices of vectors $v_1, \cdots, v_n \in R^m$, i.e. $A_{ij} = v_i^T v_j$. In other words, $A = U^T U$ for some matrix U.*
5. *A is a positive combination of matrices of the form xx^T.*
6. *The determinant of every symmetric minor of A is nonnegative.*

Let us briefly sketch the proof of some of these equivalences. Definition 3 clearly implies definition 2. In turn, definition 2 implies 4 as we can consider the eigenvector decomposition $A = V^T \Sigma V$ and let $U = \sqrt{\Sigma} V$, where condition 2 implies that $\sqrt{\Sigma}$ is well defined. Moreover, condition 4 implies condition 3 as $x^T U^T U x = ||Ux||^2 \geq 0$.

Condition 5 is equivalent as it is implied by and implies condition 3. The proof for condition 6 is slightly more involved and is known as Sylvester Criterion.

Examples Consider the following matrices.

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \qquad C = \begin{pmatrix} 3 & -2 \\ -2 & 3 \end{pmatrix}.$$

Matrix A is clearly positive semidefinite as $x^T A x = ||x||^2$. Matrix B is not positive semidefinite as $\det(B) < 0$. This is an example of a clear piece of evidence that a matrix is not positive semidefinite: if we have $D_{ii} = 0$ and $D_{ij} = D_{ji} \neq 0$,for some i, j, then $D \not\succeq 0$. This is easily seen by Sylvester Criterion as the $\{i, j\}$ minor of

D is negative. Finally, $C \succeq 0$. This is an example of a diagonally dominant symmetric matrix, i.e. a matrix D such that $D_i i \geq \sum_{j \neq i} |D_{ij}|$. Such matrices are all positive semidefinite as

$$x^T D x = \sum_i \left(D_i i x_i^2 + \sum_{j \neq i} D_{ij} x_i x_j \right) \geq \sum_i \left(\sum_{j \neq i} |D_{ij}| x_i^2 - |D_{ij}||x_i x_j| \right) \geq \sum_{i<j} |D_{ij}|(x_i^2 - 2|x_i x_j| + x_j^2) \geq 0.$$

Determining if a matrix is positive semidefinite Sylvester's criterion and the eigenvalue characterization give two ways of computationally determine whether a given matrix A is positive semidefinite. A third way is to perform a 2-sided Gaussian elimination using the diagonal elements as pivots (i.e. for every diagonal entry, perform elementary row operations to zero out the corresponding column. Then, perform elementary column operations to zero out the row). This elimination process can be performed in a particularly simple way for symmetric matrices as the row operation necessary to eliminate an entry a_{ij} can be applied to the columns to eliminate entry a_{ji}. This means that the decomposition we obtain after k steps has the form $A^{(k)} = E_1^T E_2^T \cdots E_k^T A E_k \cdots E_2 E_1$. This fact is crucial, as now we have reduced the problem of verifying $A \succeq 0$, to that of checking $A^{(k)} \succeq 0$. Notice that the decomposition may not fully diagonalize the matrix. Indeed, in the case when the next pivot of $A^{(k)}$ is a zero diagonal entry and a non-zero term on a corresponding column or row exists, it is impossible to eliminate further. However, as discussed in the examples above, this immediately implies $A^{(k)} \not\succeq 0$. Similarly, if we ever encounter a negative diagonal entry in $A^{(k)}$ we have a certficate that A is not positive semidefinite. In all other cases, the decomposition completes and shows how A is equivalent to $E^T \Sigma E$, where Σ is a nonnegative diagonal matrix. This immediately yields $A \succeq 0$.

We conclude this review of positive semidefinite matrices with two useful facts about positive semidefinite matrices. We define the matrix inner product $\cdot$ as $A \cdot B = \mathrm{Tr}(A^T B) = \sum_{i,j} A_{ij} B_{ij}$.

Lemma 1.5 *If A and B are positive semidefinite matrices, then $A \cdot B \geq 0$.*

Lemma 1.6 *A symmetric matrix A is positive semidefinite if and only if $A \cdot B \geq 0$, for every $B \succeq 0$.*

1.4.2 SDP program

We will consider two forms of semidefinite programs. They are

$$\begin{aligned} &\min \ c^T x, \\ &\text{s.t.} \ x_1 A_1 + \cdots x_n A_n - B \succeq 0, \end{aligned}$$

where $A_i \in R^{m \times m}$ and $c \in R^n$, and

$$\begin{aligned} &\max && C \cdot X, \\ &\text{s.t.} \forall i \in [k] && D_i \cdot X = d_i, \\ & && X \succeq 0, \end{aligned}$$

where C and the D_i's are symmetric matrices. These two forms can be easily shown to be equivalent.

We conclude by showing the semidefinite version of Farkas Lemma.

Lemma 1.7 *Let $A_1, \cdots A_m$ be symmetric $m \times m$ matrices. Then, $x_1A_1 + \cdots x_nA_n \succ 0$, has no solution if and only if there exists a matrix $Y \neq 0$ such that*

$$\begin{aligned} \forall i \in [n] \qquad & A_i \cdot Y = 0, \\ & Y \succeq 0. \end{aligned}$$

Proof: Interpret the symmetric $(m \times m)$-matrices as vectors and consider the cone $P = \{X : X \succeq 0\}$ and the subspace $S = \{X : \exists x \in R^n, X = \sum_i x_i A_i\}$. P and S are convex. If our semidefinte positive inequality has no solution it must be tha case that S and P have disjoint interiors. Hence, by the Hyperplane Separation Lemma, there exists a hyperplane H separating S and P. As S is a subspace, we may assume that $S \subseteq H$ and H goes through the origin, i.e. has the form $H = \{X : X \cdot Y = 0\}$ for some symmetric matrix Y. Moreover, we must have $X \cdot Y \geq 0$ for all $X \in P$. By Lemma 1.6, this implies $Y \succeq 0$. And, as $S \subseteq H$ and $A_i \in S$, we have $A_i \cdot Y = 0$ for all i. ■

In the next lecture, we will show a more general version of Farkas Lemma for inequalities of the form $x_1A_1 + \cdots x_nA_n \succ B$. We will use this to show strong duality for SDPs.

References

[1] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 222–231, New York, NY, USA, 2004. ACM.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[3] Nikhil R. Devanur, Subhash A. Khot, Rishi Saket, and Nisheeth K. Vishnoi. Integrality gaps for sparsest cut and minimum linear arrangement problems. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 537–546, New York, NY, USA, 2006. ACM.

[4] Michel X. Goemans. An introduction to linear programming. In *Combinatorial Optimization, Mathematical Programming*, pages 143–161, 1994.

[5] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103(15):150502, Oct 2009.

[6] Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. Qip = pspace. Aug 2009.

[7] Lovsz. Semidefinite programs and combinatorial optimization (lecture notes), 1995.