Lecture 3

Lecturer: Pablo A. Parrilo *Scribe: Pablo A. Parrilo*

In this lecture, we will discuss one of the most important applications of semidefinite programming, namely its use in the formulation of convex relaxations of nonconvex optimization problems. We will present the results from several different, but complementary, points of view. These will also serve us as starting points for the generalizations to be presented later in the course.

We will discuss first the case of binary quadratic optimization, since in this case the notation is simpler, and perfectly illustrates many of the issues appearing in more complicated problems. Afterwards, a more general formulation containing arbitrary linear and quadratic constraints will be presented.

1 Binary optimization

Binary (or Boolean) quadratic optimization is a classical combinatorial optimization problem. In the version we consider, we want to minimize a quadratic function, where the decision variables can only take the values ± 1. In other words, we are minimizing an (indefinite) quadratic form over the vertices of an n-dimensional hypercube. The problem is formally expressed as:

$$\begin{array}{rl} \text{minimize} & x^T Q x \\ \text{s.t.} & x_i \in \{-1, 1\} \end{array} \tag{1}$$

where $Q \in \mathcal{S}^n$. There are many well-known problems that can be naturally written in the form above. Among these, we mention the maximum cut problem (MAXCUT) discussed below, the 0-1 knapsack, the linear quadratic regulator (LQR) control problem with binary inputs, etc.

Notice that we can model the Boolean constraints using quadratic equations, i.e.,

$$x_i^2 - 1 = 0 \quad \iff \quad x_i \in \{-1, 1\}.$$

These n quadratic equations define a finite set, with an exponential number of elements, namely all the n-tuples with entries in $\{-1, 1\}$. There are exactly 2^n points in this set, so a direct enumeration approach to (1) is computationally prohibitive when n is large (already for $n = 30$, we have $2^n \approx 10^9$).

We can thus write the equivalent polynomial formulation:

$$\begin{array}{rl} \text{minimize} & x^T Q x \\ \text{s.t.} & x_i^2 = 1 \end{array} \tag{2}$$

We will denote the optimal value and optimal solution of this problem as $f_\star$ and $x_\star$, respectively. It is well-known that the decision version of this problem is *NP-complete* (e.g., [GJ79]). Notice that this is true even if the matrix Q is positive definite (i.e., $Q \succ 0$), since we can always make Q positive definite by adding to it a constant multiple of the identity (this only shifts the objective by a constant).

Example 1 (MAXCUT) *The maximum cut (MAXCUT) problem consists in finding a partition of the nodes of a graph $G = (V, E)$ into two disjoint sets V_1 and V_2 ($V_1 \cap V_2 = \emptyset, V_1 \cup V_2 = V$), in such a way to maximize the number of edges that have one endpoint in V_1 and the other in V_2. It has important practical applications, such as optimal circuit layout. The decision version of this problem (does there exist a cut with value greater than or equal to K?) is NP-complete [GJ79].*

We can easily rewrite the MAXCUT problem as a binary optimization problem. A standard formulation (for the weighted problem) is the following:

$$\max_{y_i \in \{-1,1\}} \frac{1}{4} \sum_{i,j} w_{ij}(1 - y_i y_j), \tag{3}$$

where w_{ij} is the weight corresponding to the (i,j) edge, and is zero if the nodes i and j are not connected. The constraints $y_i \in \{-1,1\}$ are equivalent to the quadratic constraints $y_i^2 = 1$.

We can easily convert the MAXCUT formulation into binary quadratic programming. Removing the constant term, and changing the sign, the original problem is clearly equivalent to:

$$\min_{y_i^2=1} \sum_{i,j} w_{ij} y_i y_j. \tag{4}$$

1.1 Semidefinite relaxations

Computing "good" solutions to the binary optimization problem given in (2) is a quite difficult task, so it is of interest to produce accurate bounds on its optimal value. As in all minimization problems, *upper bounds* can be directly obtained from feasible points. In other words, if $x_0 \in \mathbb{R}^n$ has entries equal to ± 1, it always holds that $f_\star \leq x_0^T Q x_0$ (of course, for a poorly chosen x_0, this upper bound may be very loose).

To prove *lower bounds*, we need a different technique. There are several approaches to do this, but as we will see in detail in the next sections, many of them will turn out to be exactly equivalent in the end. Indeed, many of these different approaches will yield a characterization of a lower bound in terms of the following primal-dual pair of semidefinite programming problems:

$$\begin{array}{rlcrl} \text{minimize} & \operatorname{Tr} QX & \qquad & \text{maximize} & \operatorname{Tr} \Lambda \\ \text{s.t.} & X_{ii} = 1 & & \text{s.t.} & Q \succeq \Lambda \\ & X \succeq 0 & & & \Lambda \ \text{diagonal} \end{array} \tag{5}$$

In the next sections, we will derive these SDPs several times, in a number of different ways. Let us notice here first that for this primal-dual pair of SDP, strong duality always holds, and both achieve their corresponding optimal solutions (why?).

1.2 Lagrangian duality

A general approach to obtain lower bounds on the value of general (non)convex minimization problems is to use Lagrangian duality. As we have seen the original Boolean minimization problem can be written as:

$$\begin{array}{rl} \text{minimize} & x^T Q x \\ \text{s.t.} & x_i^2 - 1 = 0 \end{array} \tag{6}$$

For notational convenience, let $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$. Then, the Lagrangian function can be written as:

$$L(x,\lambda) = x^T Q x - \sum_{i=1}^n \lambda_i (x_i^2 - 1) = x^T (Q - \Lambda) x + \operatorname{Tr} \Lambda.$$

For the dual function $g(\lambda) := \inf_x L(x,\lambda)$ to be bounded below, we need the implicit constraint that the matrix $Q - \Lambda$ must be positive semidefinite. In this case, the optimal value of x is zero, and thus we obtain a lower bound given by the solution of the SDP:

$$\begin{array}{rl} \text{maximize} & \operatorname{Tr}\Lambda \\ \text{s.t.} & Q - \Lambda \succeq 0 \end{array} \tag{7}$$

This is exactly the dual side of the SDP in (5).

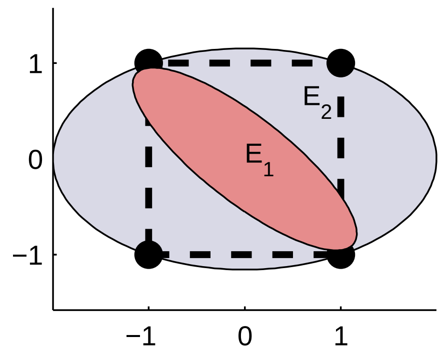

Figure 1: The ellipsoids $\mathcal{E}_1$ and $\mathcal{E}_2$.

1.3 Underestimator of the objective

A different but related interpretation of the SDP relaxation (5) is through the notion of an *underestimator* of the objective function. Indeed, the quadratic function $x^T \Lambda x$ is an "easily optimizable" function that is guaranteed to lie below the desired objective $x^T Q x$. To see this, notice that for any feasible x we have

$$x^T Q x \geq x^T \Lambda x = \sum_{i=1}^{n} \Lambda_{ii} x_i^2 = \operatorname{Tr} \Lambda,$$

where

- The first inequality follows from $Q \succeq \Lambda$
- The second equation holds since the matrix Λ is diagonal
- Finally, the third one holds since $x_i \in \{+1, -1\}$

There is also a nice corresponding geometric interpretation. For simplicity, we assume without loss of generality that Q is positive definite. Then, the problem (2) can be intepreted as finding the largest value of γ for which the ellipsoid $\{x \in \mathbb{R}^n | x^T Q x \leq \gamma\}$ does not contain a vertex of the unit hypercube.

Consider now the two ellipsoids in $\mathbb{R}^n$ defined by:

$$\mathcal{E}_1 = \{x \in \mathbb{R}^n \,|\, x^T Q x \leq \mathrm{Tr}\Lambda\}$$
$$\mathcal{E}_2 = \{x \in \mathbb{R}^n \,|\, x^T \Lambda x \leq \mathrm{Tr}\Lambda\}.$$

The principal axes of ellipsoid $\mathcal{E}_2$ are aligned with the coordinates axes (since Λ is diagonal), and furthermore its boundary contains all the vertices of the unit hypercube. Also, it is easy to see that the condition $Q \succeq \Lambda$ implies $\mathcal{E}_1 \subseteq \mathcal{E}_2$.

With these facts, it is easy to understand the related problem that the SDP relaxation is solving: dilating $\mathcal{E}_1$ as much as possible, while ensuring the existence of another ellipsoid $\mathcal{E}_2$ with coordinate-aligned axes and touching the hypercube in all 2^n vertices; see Figure 1 for an illustration.

1.4 Probabilistic interpretation

`To be written` ToDo

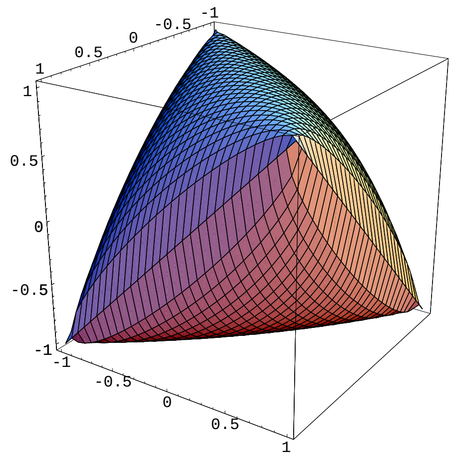

Figure 2: The three-dimensional "spectraplex." This is the set of 3×3 positive semidefinite matrices, with unit diagonal.

1.5 Lifting and rank relaxation

We present yet another derivation of the SDP relaxations, this time focused on the primal side. Recall the original formulation of the optimization problem (2). Define now $X := xx^T$. By construction, the matrix $X \in \mathcal{S}^n$ satisfies $X \succeq 0$, $X_{ii} = x_i^2 = 1$, and has *rank one*. Conversely, any matrix X with

$$X \succeq 0, \quad X_{ii} = 1, \quad \operatorname{rank} X = 1$$

necessarily has the form $X = xx^T$ for some ± 1 vector x (why?). Furthermore, by the cyclic property of the trace, we can express the objective function directly in terms of the matrix X, via:

$$x^T Q x = \operatorname{Tr} x^T Q x = \operatorname{Tr} Q x x^T = \operatorname{Tr} Q X.$$

As a consequence, the original problem (2) can be exactly rewritten as:

$$\begin{array}{rl} \text{minimize} & \operatorname{Tr} QX \\ \text{s.t.} & X_{ii} = 1, \quad X \succeq 0 \\ & \operatorname{rank}(X) = 1 \end{array}$$

This is almost an SDP problem (all the constraints are either linear or conic), except for the rank one constraint on X. Since this is a minimization problem, a lower bound on the solution can be obtained by dropping the (nonconvex) rank constraint, which enlarges the feasible set.

A useful interpretation is in terms of a nonlinear *lifting* to a higher dimensional space. Indeed, rather than solving the original problem in terms of the n-dimensional vector x, we are instead solving for the $n \times n$ matrix X, effectively converting the problem from $\mathbb{R}^n$ to $\mathcal{S}^n$ (which has dimension $\binom{n+1}{2}$).

Observe that this line of reasoning immediately shows that if we find an optimal solution X of the SDP (5) that has rank one, then we have solved the original problem. Indeed, in this case the upper and lower bounds on the solution coincide.

As a graphical illustration, in Figure 2 we depict the set of 3×3 positive semidefinite matrices of unit diagonal. The rank one matrices correspond to the four "vertices" of this convex set, and are in (two-to-one) correspondence with the eight 3-vectors with ± 1 entries.

In general, it is not the case that the optimal solution of the SDP relaxation will be rank one. However, as we will see in the next section, it is possible to use *rounding schemes* to obtain "nearby" rank one solutions. Furthermore, in some cases, it is possible to do so while obtaining some approximation guarantees on the quality of the rounded solutions.

2 Bounds: Goemans-Williamson and Nesterov

So far, our use of the SDP relaxation (5) has been limited to providing only *a posteriori* bounds on the optimal solution of the original minimization problem. However, two desirable features are missing:

- Approximation guarantees: is it possible to prove general properties on the quality of the bounds obtained by SDP?
- Feasible solutions: can we (somehow) use the SDP relaxations to provide not just bounds, but actual feasible points with good (or optimal) values of the objective?

As we will see, it turns out that both questions can be answered in the positive. As it has been shown by Goemans and Williamson [GW95] in the MAXCUT case, and Nesterov in a more general setting, we can actually achieve both of these objectives by randomly "rounding" in an appropriate manner the solution X of this relaxation. We discuss these results below.

2.1 Goemans and Williamson rounding

In their celebrated MAXCUT paper, Goemans and Williamson developed the following randomized method for finding a "good" feasible cut from the solution of the SDP.

- Factorize X as $X = V^T V$, where $V = [v_1 \dots v_n] \in \mathbb{R}^{r \times n}$, where r is the rank of X.
- Then $X_{ij} = v_i^T v_j$, and since $X_{ii} = 1$ this factorization gives n vectors v_i on the unit sphere in $\mathbb{R}^r$
- Instead of assigning either 1 or -1 to each variable, we have assigned to each a point on the unit sphere in $\mathbb{R}^r$.
- Now, choose a random hyperplane in $\mathbb{R}^r$, and assign to each variable x_i either a $+1$ or a -1, depending on which side of the hyperplane the point v_i lies.

It turns out that this procedure gives a solution that, on average, is quite close to the value of the SDP bound. We will compute the expected value of the rounded solution in a slightly different form from the original G-W argument, but one that will be helpful later. The random hyperplane can be characterized by its normal vector p, which is chosen to be uniformly distributed on the unit sphere (e.g., by suitably normalizing a standard multivariate Gaussian random variable). Then, according to the description above, the rounded solution is given by $x_i = \operatorname{sign}(p^T v_i)$. The expected value of this solution can then be written as:

$$\mathbf{E}_p[x^T Q x] = \sum_{ij} Q_{ij} \mathbf{E}_p[x_i x_j] = \sum_{ij} Q_{ij} \mathbf{E}_p[\operatorname{sign}(p^T v_i) \operatorname{sign}(p^T v_j)].$$

We can easily compute the value of this expectation. Consider the plane spanned by v_i and v_j, and let θ_{ij} be the angle between these two vectors. Then, it is easy to see that the desired expectation is equal to the probability that both points are on the same side of the hyperplane, minus the probability that they are on different sides. These probabilities are $1 - \frac{\theta_{ij}}{\pi}$ and $\frac{\theta_{ij}}{\pi}$, respectively. Thus, the expected value of the rounded solution is exactly:

$$\sum_{ij} Q_{ij} \left(1 - \frac{2\theta_{ij}}{\pi}\right) = \sum_{ij} Q_{ij} \left(1 - \frac{2}{\pi} \arccos(v_i^T v_j)\right) = \frac{2}{\pi} \sum_{ij} Q_{ij} \arcsin X_{ij}. \tag{8}$$

Notice that the expression is of course well-defined, since if X is PSD and has unit diagonal, all its entries are bounded in absolute value by 1. This result exactly characterizes the expected value of the rounding procedure, as a function of the optimal solution of the SDP. We would like, however, to directly relate this quantity to the optimal solution of the original optimization problem. For this, we will need additional assumptions on the matrix Q. We discuss next two of the most important results in this direction.

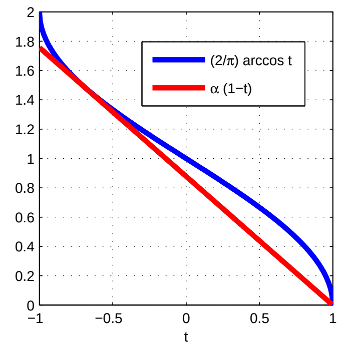

Figure 3: Bound on the inverse cosine function, for $\alpha \approx 0.878$.

2.2 MAXCUT bound

Recall from (3) that for the MAXCUT problem, the objective function does not only include the quadratic part, but there is actually a constant term:

$$\frac{1}{4}\sum_{ij} w_{ij}\,(1 - y_i y_j).$$

The expected value of the cut is then:

$$c_{\text{sdp-expected}} = \frac{1}{4}\sum_{ij} w_{ij}\left(1 - \frac{2}{\pi}\arcsin X_{ij}\right) = \frac{1}{4}\cdot\frac{2}{\pi}\sum_{ij} w_{ij}\arccos X_{ij}.$$

On the other hand, the solution of the SDP gives an upper bound on the cut capacity equal to:

$$c_{\text{sdp-upper-bound}} = \frac{1}{4}\sum_{ij} w_{ij}(1 - X_{ij}).$$

To relate these two quantities, we look for a constant α such that

$$\alpha\,(1-t) \leq \frac{2}{\pi}\arccos(t) \qquad \text{for all } t \in [-1,1]$$

The best possible (i.e., largest) such constant is $\alpha = 0.878$; see Figure 3. So we have

$$c_{\text{sdp-upper-bound}} \leq \frac{1}{\alpha}\cdot\frac{1}{4}\cdot\frac{2}{\pi}\sum_{ij} w_{ij}\arccos X_{ij} = \frac{1}{\alpha}\,c_{\text{sdp-expected}}$$

Notice that here we have used the nonnegativity of the weights (i.e., $w_{ij} \geq 0$). Thus, so far we have the following inequalities:

- $c_{\text{sdp-upper-bound}} \leq \frac{1}{\alpha}\,c_{\text{sdp-expected}}$
- Also clearly $c_{\text{sdp-expected}} \leq c_{\max}$
- And $c_{\max} \leq c_{\text{sdp-upper-bound}}$

Putting it all together, we can sandwich the value of the relaxation as follows:

$$\boxed{\alpha\cdot c_{\text{sdp-upper-bound}} \leq c_{\text{sdp-expected}} \leq c_{\max} \leq c_{\text{sdp-upper-bound}}.}$$

2.3 Nesterov's $\frac{2}{\pi}$ result

A result by Nesterov generalizes the MAXCUT bound described above, but for a larger class of problems. The original formulation is for the case of binary *maximization*, and applies to the case when the matrix A is *positive semidefinite*. Since the problem is homogeneous, the optimal value is guaranteed to be nonnegative.

As we have seen, the expected value of the solution after randomized rounding is given by (8). Since X is positive semidefinite, it follows from the nonnegativity of the Taylor series of $\arcsin(t) - t$ and the Schur product theorem that

$$\arcsin[X] \succeq X,$$

where the arcsin function is applied componentwise. This inequality can be combined with (8) to give the bounds:

$$\boxed{\frac{2}{\pi} \cdot f_{\text{sdp-upper-bound}} \leq f_{\text{sdp-expected}} \leq f_{\max} \leq f_{\text{sdp-upper-bound}},}$$

where $2/\pi \approx 0.636$. For more details, see [BTN01, Section 4.3.4]. Among others, the paper [Meg01] presents several new results, as well as a review of many of the available approximation schemes.

3 Linearly constrained problems

In this section we extend the earlier results, to general quadratic optimization problems under linear and quadratic constraints. For notational simplicity, we write the constraints in homogeneous form, i.e., in terms of the vector $x = \begin{bmatrix} 1 & y^T \end{bmatrix}^T$.

The general primal form of the SDP optimization problems we are concerned with is

$$\begin{aligned} \min \quad & x^T Q x \\ \text{s.t.} \quad & x^T A_i x \geq 0 \\ & Bx \geq 0 \\ & x = \begin{bmatrix} 1 \\ y \end{bmatrix} \end{aligned}$$

The corresponding primal and dual SDP relaxations are given by

$$\boxed{\begin{aligned} \min \quad & Q \bullet X \\ \text{s.t.} \quad & A_i \bullet X \geq 0 \\ & BXB^T \geq 0 \\ & E_{11} \bullet X = 1 \\ & X \succeq 0 \end{aligned} \qquad \begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & Q \succeq \gamma\, E_{11} + \textstyle\sum_i \lambda_i\, A_i + B^T N B \\ & \lambda_i \geq 0 \\ & N \geq 0 \\ & N_{ii} = 0 \end{aligned} \qquad (9)}$$

Here E_{11} denotes the matrix with a 1 on the $(1, 1)$ component, and all the rest being zero. The dual variables λ_i can be interpreted as Lagrange multipliers associated to the quadratic constraints of the primal problem, while N corresponds to pairwise products of the linear constraints.

References

[BTN01] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization.* MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the theory of NP-completeness*. W. H. Freeman and Company, 1979.

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[Meg01] A. Megretski. Relaxations of quadratic programs in operator theory and system analysis. In *Systems, approximation, singular integral operators, and related topics (Bordeaux, 2000)*, volume 129 of *Oper. Theory Adv. Appl.*, pages 365–392. Birkhäuser, Basel, 2001.