Lecture 5

Strong duality for SDPs
Matrix multiplicative weights algorithm*

5.1 Strong duality theorem for SDPs

First we will prove the strong duality theorem for SDPs. Recall the canonical forms of SDP:

\[(P) : \sup B \cdot X \quad \text{s.t.} \quad A_i \cdot X = c_i \quad \forall i = 1 \ldots m \]
\[X \succeq 0 \quad (5.1)\]

\[(D) : \inf c^\top y \quad \text{s.t.} \quad \sum_i y_i A_i - B \succeq 0 \quad (5.2)\]

We will sometimes find it more convenient to consider an alternative form, where the primal constraints are relaxed to inequalities and the dual constraints include \(y_i \geq 0\):

\[(P) : \sup B \cdot X \quad \text{s.t.} \quad A_i \cdot X \leq c_i \quad \forall i = 1 \ldots m \]
\[X \succeq 0 \quad (5.3)\]

\[(D) : \inf c^\top y \quad \text{s.t.} \quad \sum_i y_i A_i - B \succeq 0 \]
\[y_i \geq 0 \quad (5.4)\]

*Lecturer: Thomas Vidick. Scribe: Hoang M. Le.*
**Theorem 5.1** (Strong duality). Suppose both (P) and (D) are strictly feasible and bounded. Let $\alpha$ and $\beta$ be their optimal values. Then $\alpha = \beta$.

**Proof.** Suppose for contradiction that $\alpha < \beta$. Let $\alpha' = \frac{\alpha + \beta}{2}$ be the midpoint. Define the set

$$K = \left\{ \left( \sum_i y_i A_i - B, \alpha' - c^\top y \right) \in \mathbb{R}^{n+1}, y \in \mathbb{R}^m \right\}$$

Then $K$ does not intersect the positive cone $\text{Pos}(\mathbb{R}^n) \times \mathbb{R}_+$. By definition $K$ is an affine subspace, hence there exists an affine hyperplane \( \left( \begin{array}{c} Z \\ \mu \end{array} \right) \in \mathbb{R}^{n+1}, \delta \in \mathbb{R} \) that includes $K$ but does not intersect $\text{Pos}(\mathbb{R}^n) \times \mathbb{R}_+$:

- $\forall y \in \mathbb{R}^m, Z \cdot (\sum y_i A_i - B) + \mu(\alpha' - c^\top y) = \delta$
- $\forall Q \succeq 0, \forall q \geq 0, Z \cdot Q + \mu q > \delta$

Taking $Q = 0, q = 0$ in the second constraint shows that necessarily $\delta \leq 0$. Suppose $Z$ has a negative eigenvalue. For any value of $\delta$, taking $q = 0$ and $Q$ as a large enough multiple of the projection on the eigenvector of $Z$ corresponding to this eigenvalue would contradict the second inequality. Thus $Z \succeq 0$.

Setting $y = 0$ in the first constraint gives $Z \cdot B = \alpha' \mu - \delta$. Setting $y = e_i$, we obtain $Z \cdot A_i = \mu c_i$. Finally, taking $y$ to be any strictly feasible point, meaning $\left\{ \sum_i y_i A_i - B \succ 0 \right\}$ and $\left\{ \alpha' - c^\top y < 0 \right\}$ (since any feasible solution will satisfy $c^\top y \geq \beta \succ \alpha'$), gives $\mu > 0$.

Define $Z' = \frac{Z}{\mu}$. Then $Z'$ is primal feasible: $Z' \succeq 0$ since both $Z \succeq 0$ and $\mu > 0$, and $Z' \cdot A_i = c_i$ for each $i$. We can also compute the objective value $Z' \cdot B = \alpha' - \frac{\delta}{\mu} > \alpha$ since $\delta < 0, \mu > 0$. This is a contradiction with our original assumption that the primal optimum was $\alpha$.

### 5.2 Matrix Multiplicative Weights Algorithm

We will see how a large class of semidefinite programs can be solved very efficiently using an extension of the Multiplicative Weights Algorithm to the case of matrices. Recall the set-up of the algorithm:

- We have $n$ experts
- At each step $t = 1, \cdots, T$, the player chooses a distribution $p^{(t)}$ over experts
- A cost vector $m^{(t)}$ is supplied by the environment
- The player suffers a loss of $p^{(t)} \cdot m^{(t)}$
Now suppose there is a continuous set of experts, each associated with a unit vector $v \in \mathbb{R}^n, \|v\| = 1$. Let the distribution over experts be a distribution $D$ on the set of all unit vectors, the unit sphere $\mathcal{S}^{n-1}$. Additionally let the loss be specified by a matrix $M \in \mathbb{R}^{n \times n}, 0 \preceq M \preceq I$ ($M$ is symmetric) such that by definition the loss associated with expert $v$ is given by $v^\top M v$. Our assumption on $M$ implies this loss will always fall in the interval $[0, 1]$. The expected loss over distribution $D$ is given by:

$$
\mathbb{E}_{v \sim D}[v^\top M v] = \mathbb{E}_{v \sim D}[M \cdot vv^\top] = M \cdot \underbrace{\mathbb{E}_{v \sim D}[vv^\top]}_{\rho \geq 0},
$$

where $\rho \geq 0$ satisfies $\text{Tr}(\rho) = \mathbb{E}_{v \sim D}\|v\|^2 = 1$. $\rho$ is called density matrix. (As a notation reminder, $A \cdot B = \text{Tr}(A^\top B)$).

We then have the following extension of MWA:

**Definition 5.2.** (MMWA) Let $\eta > 0$. Initialize $X^{(0)} = \frac{1}{n}$. For $t = 1, \ldots, T$:

- Observe cost matrix $0 \preceq M^{(t)} \preceq I$
- Set $W^{(t)} = \exp(-\eta(M^{(1)} + \cdots + M^{(t)}))$.
- Set $X^{(t)} = \frac{W^{(t)}}{\text{Tr}(W^{(t)})}$

**Remark 5.3.** Note that loss is given by the inner product between $X^{(t)}$ and $M^{(t)}$ ($X^{(t)} \cdot M^{(t)} = \text{Tr}(X^{(t)} \cdot M^{(t)})$), and that $0 \leq \text{Tr}(X^{(t)} \cdot M^{(t)}) \leq \text{Tr}(X^{(t)} \cdot I) = 1$.

**Remark 5.4.** This is a strict generalization of MWA. We can recover the experts framework by considering the special case where all loss matrices are diagonal:

$$
M^{(t)} = \begin{pmatrix}
m_1^{(t)} & 0 \\
0 & \ddots \\
0 & \cdots & m_n^{(t)}
\end{pmatrix}
$$

**Remark 5.5.** There are many equivalent definitions of the exponential of a symmetric matrix $A$. A first definition goes through diagonalization of $A$. Specifically, let $A = UDU^\top$, where $U$ is orthogonal and $D$ is diagonal. Then one can define $\exp(A) = U \exp(D) U^\top =$

$$
U \begin{pmatrix}
e^{D_{11}} & 0 \\
0 & \ddots \\
0 & \cdots & e^{D_{nn}}
\end{pmatrix} U^\top.
$$

3
More generally, for any function \( f \) we can define \( f(A) \) in this way. Taking \( f(x) = x^k \) for some integer \( k \), you can verify that this definition agrees with the usual definition of the matrix product. For instance, \( A^2 = (UDU^T)(UDU^T) = U D^2 U^T \) since \( U \) is orthogonal.

Another efficient, more efficient for practical computation, goes through the Taylor expansion of the exponential: \( \exp(A) = I + A + \frac{A^2}{2!} + \cdots + \frac{A^k}{k!} + \cdots \), which converges exponentially fast.

Remark 5.6. Note that for matrices, \( \exp(A + B) \neq \exp(A)\exp(B) \) in general. Thus the definition of \( W(t) \) given above is not equivalent in general to the update rule \( W(t) := W(t-1)\exp(-\eta M(t)) \) that we saw in the experts framework; the two rules are identical only if all matrices \( M(t) \) commute. In general, we only know the inequality \( \text{Tr}(e^{A+B}) \leq \text{Tr}(e^Ae^B) \), which is known as the Golden-Thompson inequality.

The following theorem is a direct extension of the experts theorem:

**Theorem 5.7.** For any \( M^{(1)}, \ldots, M^{(T)} \in \mathbb{R}^{n \times n} \) such that \( 0 \preceq M^{(i)} \preceq I \) for all \( i \), and \( \epsilon > 0 \), let \( \eta = -\ln(1-\epsilon) \). Then for any \( v \in \mathbb{R}^n, \|v\| = 1 \), the following relation holds for \( M \) and \( X \) defined via Matrix Multiplicative Weights Algorithms with parameter \( \eta \):

\[
\sum_{t=1}^{T} M^{(t)} \cdot X^{(t)} \leq (1 + \epsilon) \sum_{t=1}^{T} M^{(t)} \cdot vv^\top + \frac{\ln n}{\epsilon}.
\]  

**Proof.** Homework (will need to use Golden-Thompson inequality and requires some modification of the potential function). 

\( \square \)

### 5.3 Application of Matrix Multiplicative Weights Algorithm

We want to use the algorithm presented above to solve SDPs. First we reduce the optimization problem to a feasibility decision problem. Suppose we are given the optimization problem in standard form:

\[
(P) : \sup B \cdot X \\
\text{s.t. } A_i \cdot X \leq c_i \quad \forall i = 1 \ldots m \\
X \succeq 0
\]

\[
(D) : \inf c^\top y \\
\text{s.t. } \sum_i y_i A_i - B \succeq 0 \\
y_i \geq 0
\]
Assume $A_1 = I, c_1 = R$, which gives the constraint that $\text{Tr}(X) \leq R$, effectively imposing an a priori bound on the size of $X$. Assume also that the optimal $\alpha$ of the primal problem is non-negative (if not, we can always shift the optimum by adding a positive multiple of the identity to $B$ to make it PSD). To perform a reduction from deciding optimality to deciding feasibility we perform a binary search over $\alpha \in [0, \|B\| R]$ (where $\|B\|$ is the largest eigenvalue of $B$, so $|B \cdot X| \leq \|B\| \text{Tr}(X) \leq BR$, at each step deciding feasibility of the following problem:

$$\exists ? X \quad \text{s.t.} \quad B \cdot X > \alpha$$
$$A_i \cdot X \leq c_i$$
$$X \succeq 0$$

(5.6)

We will use the algorithm presented in the previous section to answer this decision problem. The idea is to start with a benign guess of $X$, such as $X = R I / n$. Then we will iteratively “improve” $X$ to either make it into a feasible point, thus answering our feasibility problem, or somehow obtain a proof that this is not possible, in which case we know $\alpha$ is not reachable by the original SDP. If $\alpha$ is found to be unreachable we repeat our search at $0$ and $\alpha$; if it is feasible we iterate between $\alpha$ and $\|B\| R$. The number of iteration will be logarithmic in terms of the bound $\|B\| R$.

We now state the main claim that underlies our “improvement” subroutine for $X$:

Claim 5.8. The following are equivalent:

- (i) $\exists y \in \mathbb{R}^n, y \succeq 0, c^\top y \leq \alpha$ such that $X \cdot (\sum_i y_i A_i - B) \geq 0$
- (ii) either $X$ is infeasible or $B \cdot X \leq \alpha$

Proof. Assume (i), suppose $X$ is feasible, we want to show that $B \cdot X \leq \alpha$. From the assumption in (i), we have that: $0 \leq \sum_i y_i A_i \cdot X - X \cdot B \leq y^\top c - X \cdot B \leq \alpha - X \cdot B$. Thus $X \cdot B \leq \alpha$.

Conversely, assume (i) does not hold. Reasoning by contrapositive, we want to show (ii) does not hold.

We can assume that $B \cdot X > 0$, as otherwise taking $y = 0$ shows that (i) in fact holds. Scale $X$ so that we have $B \cdot X > \alpha$. Consider $y = \frac{\alpha}{c_i} e_i$ for $i = 1, \cdots, m$. We then have $y$ satisfies $y \succeq 0$, and $c^\top y = \alpha$. Using our assumption that (i) does not hold, it must be that $X \cdot (\sum_i y_i A_i - B) < 0$, thus $X \cdot A_i \frac{\alpha}{c_i} < X \cdot B = \alpha$. So $\forall i = 1, \cdots, m$, we have $X \cdot A_i < c_i$. Hence $X$ is feasible with objective value strictly larger than $\alpha$, so (ii) does not hold, as required.

We conclude that (i) and (ii) are equivalent. 

The equivalence stated in the claim shows that the existence of a $y$ such that condition (i) is satisfied is a proof that we have not yet reached our goal of finding a good feasible solution. The idea is that solving (i) is much easier than solving the SDP. Note that $y$ does
not need to be dual feasible and in fact, (i) is an LP.

Let’s make an assumption and under this assumption we’ll be able to solve the SDP.

**Assumption:** There exists an oracle $O$ such that given $X$ as an input, $O$ returns either:

- (a) a vector $y$ such that (i) in the claim above holds or
- (b) a statement that no such $y$ exists

We measure the quality of $O$ through its “width” $\sigma$, which is the largest possible value that $\| \sum_i y_i A_i - B \|$ can take over all vectors $y$ that the oracle might return. When designing an oracle, we want it to be fast and have small width. Given such an oracle, we have the following algorithm:

**Algorithm for solving SDP:** Let $X^{(1)} = R I_n, \eta = \frac{\delta \alpha^2}{2\sigma R}$, $\eta' = -\ln(1 - \eta)$, where $\delta > 0$ is a measure of accuracy.

For $t = 1, \cdots, T$:

1. Run $O$ on $X^{(t)}$. If case (b) under above assumption happens, then using the claim (i) does not hold, meaning (ii) does not either, hence $X$ is feasible and $B \cdot X > \alpha$, so we are done. Stop and return $X^{(t)}$.
2. If instead case (a) holds we update $X$ as follows. Let $y^{(t)}$ be returned by $O$. Set the loss matrix
   
   $$M^{(t)} = \sum_i y_i^{(t)} A_i - B + \sigma I_{2\sigma},$$

   so that the assumption on the width of $O$ implies $0 \leq M^{(t)} \leq I_n$.
3. Update $X^{(t+1)}$ as in MMWA:
   
   $$W^{(t+1)} = \exp(-\eta' \sum_{i=1}^t M^{(i)})$$
   
   $$X^{(t+1)} = \frac{RW^{(t+1)}}{\text{Tr}(W^{(t+1)})}.$$ 

The following theorem states the guarantee for the above algorithm:

**Theorem 5.9.** Assume $O$ does not fail for $T = \frac{8\sigma^2 R^2}{\delta^2 \alpha^2} \ln(n)$ steps. Then $\bar{y} = \frac{\delta \alpha}{R} e_1 + \frac{1}{T} \sum_{i=1}^T y^{(t)}$ is dual feasible with objective value less than or equal to $(1 + \delta)\alpha$.

**Proof.** Proof will be given in the next lecture. 

Comments on running time of the algorithm:

- Performance depends on the running time of the oracle.
- We need to compute matrix exponential to update $X$ and it is important to do this efficiently.
• We would like to have \( R \) and the width of the oracle to be not too large, otherwise the algorithm will require lots of iterations. We will see how to do this for the special case of the MAXCUT SDP in the next lecture.

• The algorithm depends inverse polynomially on the approximation error \( \delta \alpha \). This is not great, and in general we could hope for an algorithm that depends only polylogarithmically on \( \delta^{-1} \). For example this is the case for the ellipsoid algorithm.

• If all these parameters, \( R, \sigma, \delta \alpha \) are constants, or such that \( \sigma R/(\delta \alpha) \) is not too large, then the overhead \( \ln(n) \) in the number of iterations for the algorithm is very small. This is the main strength of the MMWA, and we will see how to take advantage of it for the case of MAXCUT.