# CSC2411 - Linear Programming and Combinatorial Optimization*
# Lecture 10: Semidefinite Programming

Notes taken by Mike Jamieson

March 28, 2005

**Summary:** In this lecture, we introduce semidefinite programming (SDP). We describe SDP as a generalization of linear programming (LP), and how an LP and other problems can be written as an SDP. We present vector programming (VP) as another form of SDP and discuss duality in the SDP context. Finally, we note some important differences between LP and SDP.

## 1 Components of Semidefinite Programming

Today we go beyond LP into semidefinite programming (SDP). The development of SDP was one of the major successes of the 1990's. It is a special case of convex optimization with many useful properties.

### 1.1 LP and Closed Cones

Recall that in standard form, and LP may be written as

$$\begin{array}{ll} \min\langle c, x\rangle & \text{such that} \\ \langle a_i, x\rangle = b_i, & i = 1 \ldots m \\ x \geq 0 & \end{array}$$

If we define $K = (\mathbb{R}^+)^n = \{x | x \geq 0\}$ then we can rewrite the last condition above as $x \in K$.

**Definition 1.1.** A set $K \in \mathbb{R}^n$ is a *closed cone* if $\forall x, y \in K$ and $\forall \alpha, \beta \in \mathbb{R}^+$

$$\alpha x + \beta y \in K$$

and $K$ is *closed*.

* Lecture Notes for a course given by Avner Magen, Dept. of Computer Science, University of Toronto.

## 1.2 Positive Semidefinite Matrices

**Definition 1.2.** An $n \times n$ matrix $X$ is *positive semidefinite* (PSD) if

$$\forall a \in \mathbb{R}^n, a^t X a \geq 0$$

We can denote the set of all symmetric $n \times n$ PSD matrices as $PSD_n$

$$PSD_n = \{X | X \text{ is an } n \times n \text{ symmetric matrix}, X \text{ is PSD}\}$$

**Claim 1.3.** *$PSD_n$ is a closed clone.*

*Proof.* To see that $PSD_n$ is closed under positive addition, consider that $\forall X, Y \in PSD_n$ and $\forall \alpha, \beta \in \mathbb{R}^+$

$$a^t(\alpha X + \beta Y)a = \alpha a^t X a + \beta a^t Y a \geq 0$$

To show that $PSD_n$ to be closed, it is sufficient to show that its compliment $\overline{PSD}_n$ is open. For every $X \in \overline{PSD}_n$ and any $Y \in \mathbb{R}^{n \times n}$ (within finite elements), there exists $a \in \mathbb{R}^n$ and $\varepsilon, k > 0$ sufficiently small that

$$\begin{aligned} a^t X a + \varepsilon &< 0 \\ k a^t Y a &< \varepsilon \\ a^t(X + kY)a &< 0 \end{aligned}$$

This illustrates that for any $X \in \overline{PSD}_n$ any sufficiently small perturbation is also in $\overline{PSD}_n$. Therefore, $\overline{PSD}_n$ is open and its compliment $PSD_n$ is closed. □

In the LP formulation, we require $x \in (\mathbb{R}^+)^n$. In the next section, we describe semidefinite programs in which the cone $(\mathbb{R}^+)^n$ is replaced by $PSD_n$. For now, we concentrate on the algebraic properties of $PSD_n$.

**Claim 1.4.** *For a symmetric matrix $A$, the following are equivalent*

(i) $A$ is $PSD$

(ii) the eigenvalues of $A$ are non-negative

(iii) $A$ can be written as $V^t V$ ($V$ is an $n \times n$ matrix)

*Proof.* To show $(iii) \to (i)$, consider that, given $(iii)$,

$$\begin{gathered} \forall V \in \mathbb{R}^{n \times n}, \forall x \in \mathbb{R}^n, \\ x^t A x = x^t V^t V x = \|Vx\|_2^2 \geq 0 \end{gathered}$$

For $(i) \to (ii)$, let $\lambda \in \mathbb{R}$ be an eigenvalue of $A$ and $x \in \mathbb{R}^n$ the corresponding eigenvector, such that $Ax = \lambda x$. Then $\forall A \in PSD_n$,

$$\lambda \|x\|_2^2 = x^t \lambda x = x^t A x \geq 0$$

To demonstrate $(ii) \to (iii)$, we employ the fact that if $A$ is an $n \times n$ symmetric matrix, it has a decomposition

$$A = U^t D U$$

where $U$ and $D$ are $n \times n$ matrices, $U$ is orthogonal, $D$ is diagonal and the diagonal elements of $D$ are the eigenvalues of $A$. If all eigenvalues of $A$ are non-negative, $\sqrt{D} \in \mathbb{R}^{n \times n}$, so

$$A = U^t D U = (U^t \sqrt{D}^t)(\sqrt{D} U) = V^t V$$

$\square$

**Remark 1.5.** To sketch how we can decompose $A$, we note that a symmetric matrix is made diagonal using symmetric row and column operations. One row operation can remove a term below the main diagonal while the symmetric column operation removes the corresponding above-diagonal term.

$$\begin{pmatrix} 1 & 2 & \cdots \\ 2 & 7 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad \overset{R_2 = R_2 - 2R_1}{\Longrightarrow} \quad \begin{pmatrix} 1 & 2 & \cdots \\ 0 & 3 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad \overset{C2 = C_2 - 2C_1}{\Longrightarrow} \quad \begin{pmatrix} 1 & 0 & \cdots \\ 0 & 3 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

These row and column operations are equivalent to modifying $I$ left and right and applying to $A$

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ -2 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} A \begin{pmatrix} 1 & -2 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

**Remark 1.6.** $(i) \to (iii)$ says that $A$ is PSD if and only if $\exists v_1, \ldots, v_n \in \mathbb{R}^n$ so that $a_{ij} = \langle v_i, v_j \rangle$

$$A \in PSD \Leftrightarrow A = (V^t)(V) = \begin{pmatrix} -v_1- \\ \vdots \\ -v_n- \end{pmatrix} \begin{pmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{pmatrix} = (\langle v_i, v_j \rangle)_{ij}$$

So each element $a_{ij}$ encodes the dot product of a pair of vectors, and $A$ as a whole can be thought of as encoding the relationships among a set of vectors. Note that $V$ is not unique. We can rotate or reflect the set of vectors without changing the inner products.

**Remark 1.7.** For $x \in (\mathbb{R}^+)^n$, all elements of $x$ are non-negative. $X \in PSD_n$ does not place such a constraint on the individual elements. Instead, from $(i) \to (ii)$, we see that the eigenvalues of the matrix as a whole are non-negative.

**Example 1.8.** Some instances of PSD matrices $A$ with the corresponding $V$ and $v_i$.

(i) Let $a_1, \dots, a_n \geq 0$, $A = \text{diag}(a_1, \dots, a_n)$

$$A = \begin{pmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_n \end{pmatrix} \quad V = \begin{pmatrix} \sqrt{a_1} & & 0 \\ & \ddots & \\ 0 & & \sqrt{a_n} \end{pmatrix} \quad v_i = \begin{pmatrix} \vdots \\ 0 \\ \sqrt{a_i} \\ 0 \\ \vdots \end{pmatrix}$$

(ii)

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \quad V = \begin{pmatrix} 1 & -1 \\ 1 & 0 \end{pmatrix} \quad v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad v_2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

### 1.3 Matrix Inner Product (Frobenius Inner Product)

We would like to view matrices as vectors when we apply linear constraints, and optimize linear functions on them. The matrix inner product is SDP's counterpart to the vector inner product in LP.

**Definition 1.9.** The *matrix inner product* of $A$ and $B$ is

$$A \bullet B = \sum_{\forall i,j} a_{ij} b_{ij} = \text{trace}(AB^t)$$

Notice that if $A$ and $B$ are thought of simply as reshaped vectors, this definition is equivalent to the usual vector inner product.

## 2 Writing Semidefinite Programs

An *Semidefinite Program* (SDP) has the form

$$\begin{array}{cc} \min C \bullet X & \text{such that} \\ A_i \bullet X = b_i, & i = 1 \dots m \\ X \succeq 0 & \end{array}$$

where $X$, $C$ and each $A_i$ are $n \times n$ matrices. Whereas in LP we had $n$ variables, in SDP the objective, constraints and solution matrices have $n^2$ elements. Note that we may also write SDPs where the objective is to maximize $C \bullet X$. The last condition, $X \succeq 0$, is an order relation in which $0$ represents the $n \times n$ zero matrix. $X \succeq 0$ means that $X \in PSD_n$.

$$X \succeq B \Leftrightarrow B \preceq X \Leftrightarrow (X - B) \succeq 0 \Leftrightarrow \forall x \in \mathbb{R}^n, x^t(X - B)x \geq 0$$

Next we study some examples of semidefinite programs.

**Example 2.1.** $X$ is $n \times n$ where $n = 3$, and there are $m = 2$ constraints.

$$C = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad A_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 7 \\ 1 & 7 & 0 \end{pmatrix} \quad A_2 = \begin{pmatrix} 0 & 2 & 0 \\ 2 & 6 & 0 \\ 0 & 0 & 4 \end{pmatrix} \qquad \begin{array}{rcr} b_1 & = & 0 \\ b_2 & = & -2 \end{array}$$

Writing the problem more compactly in terms of the elements of $X = (x_{ij})$, we have

$$\begin{gathered} \min \textstyle\sum_{i,j} x_{ij} \\ x_{11} + 2x_{13} + 14x_{23} = 0 \\ 4x_{12} + 6x_{22} + 4x_{33} = -2 \\ X \in PSD_n \end{gathered}$$

**Example 2.2.** Given the following incomplete matrix,

$$\begin{pmatrix} 5 & ? & ? \\ -3 & 4 & ? \\ ? & ? & ? \end{pmatrix}$$

We want to complete to a PSD matrix while minimizing the sum of entries. Representing the solution matrix as $X = (x_{ij})$, we write

$$\begin{gathered} \min \textstyle\sum_{i,j} x_{ij} \;\; \text{s.t.} \\ x_{11} = 5,\; x_{21} = -3,\; x_{22} = 4 \\ X \in PSD_n \end{gathered}$$

Expressed in matrix form,

$$\min \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \bullet X \qquad\qquad \text{s.t.}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X = 5, \quad \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X = 3, \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X = 4$$

$$X \succeq 0$$

**Observation 2.3.** *LP is a special case of SDP*

It is enough to consider the standard form:

$$
\begin{array}{ccc}
\text{LP} & \Longrightarrow & \text{SDP} \\[1em]
\min\langle c, x\rangle & \Longrightarrow & \min \begin{pmatrix} c_1 & & 0 \\ & \ddots & \\ 0 & & c_n \end{pmatrix} \bullet X \\[1em]
x \text{ has } n \text{ elements} & \Longrightarrow & \begin{array}{c} \forall i \neq j, X_{ij} = 0 \\ \text{(by inner product constraints)} \end{array} \\[1em]
i = 1 \dots m, \langle a_i, x\rangle = b_i & \Longrightarrow & i = 1 \dots m, \begin{pmatrix} a_i^{(1)} & & 0 \\ & \ddots & \\ 0 & & a_i^{(n)} \end{pmatrix} \bullet X = b_i
\end{array}
$$

While we would not normally choose to write an LP in this form due to its inefficiency, it is useful to keep in mind that SDP is an extension of LP. Unfortunately, as we shall see later, not all the nice properties of LP are maintained in the extension.

**Example 2.4.** Suppose you have two sets of points in $\mathbb{R}^n$. $\mathcal{P} = \{p_1, \dots, p_r\}$, and $\mathcal{Q} = \{q_1, \dots, q_s\}$. In Figure 1, points in $\mathcal{P}$ are represented by o's and points in $\mathcal{Q}$ are represented by x's. We wish to find an ellipse, centred at the origin, that includes all $p_i \in \mathcal{P}$ and excludes all $q_i \in \mathcal{Q}$ (where we allow $q_i$ to lie on the boundary). Recall that an ellipse centred at the origin is $\{x | x^t A x \leq 1\}$ for some $A \in PSD_n$. Here we use the $n \times n$ matrix $X \in PSD_n$ to represent the ellipse,

$$
\begin{array}{c}
\forall p_i \in \mathcal{P}, p_i^t X p_i \leq 1 \\
\forall q_i \in \mathcal{Q}, q_i^t X q_i \geq 1
\end{array} \tag{1}
$$

We write the ellipse constraints (1) as matrix inner products using, $p^t X p = (pp^t) \bullet X$, where $(pp^t)$ is the outer product (which also happens to be PSD). Finally, we add slack variables $s_{p_i}$ and $s_{q_i}$ to transform each constraint into an equality.

$$
\begin{array}{ccccc}
\forall p_i \in \mathcal{P}, p_i^t X p_i \leq 1 & \Leftrightarrow & (p_i p_i^t) \bullet X \leq 1 & \Rightarrow & (p_i p_i^t) \bullet X + s_{p_i} = 1 \\[0.5em]
\forall q_i \in \mathcal{Q}, q_i^t X q_i \geq 1 & \Leftrightarrow & (q_i q_i^t) \bullet X \geq 1 & \Rightarrow & (q_i q_i^t) \bullet X - s_{q_i} = 1
\end{array}
$$

We can incorporate the slack variables by writing the augmented matrix $X'$ as a block-diagonal

$$
X' = \begin{pmatrix} X & 0 & 0 \\ 0 & S_{\mathcal{P}} & 0 \\ 0 & 0 & S_{\mathcal{Q}} \end{pmatrix} \quad S_{\mathcal{P}} = \begin{pmatrix} s_{\mathcal{P}_1} & & 0 \\ & \ddots & \\ 0 & & s_{\mathcal{P}_r} \end{pmatrix} \quad S_{\mathcal{Q}} = \begin{pmatrix} s_{\mathcal{Q}_1} & & 0 \\ & \ddots & \\ 0 & & s_{\mathcal{Q}_s} \end{pmatrix}
$$

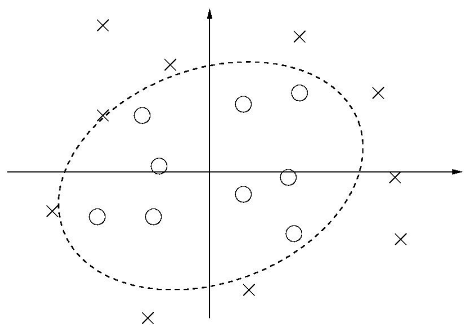

Figure 1: An ellipse centred at the origin, separating two sets of points.

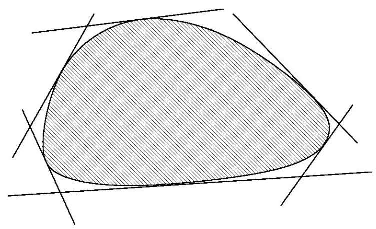

Figure 2: A feasible region bounded by an infinite family of linear constraints.

Then $\forall p_i \in \mathcal{P}$ we have a constraint in matrix form:

$$\begin{pmatrix} p_i p_i^t & 0 & 0 \\ 0 & E_i & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X' = 1$$

where $E_i$ is an $n \times n$ matrix with zeros everywhere except the $i^{th}$ position along the diagonal which is one. Similar constraints may be written $\forall q_i \in \mathcal{Q}$. Note that the final condition $X' \succeq 0$ ensures simultaneously that $X \succeq 0$ and that $S_{\mathcal{P}}, S_{\mathcal{Q}} \succeq 0$. This in turn ensures that all slack variables are non-negative.

**Remark 2.5.** In the example 2.4, we rewrote the PSD property in terms of the outer product of a vector with itself and the matrix inner product,

$$X \in PSD \Leftrightarrow \forall a, a^t X a \geq 0 \Leftrightarrow \forall a, (aa^t) \bullet X \geq 0$$

For a specific vector, $a$, the constraint $(aa^t) \bullet X \geq 0$ acts like a single LP constraint, a hyperplane in $\mathbb{R}^{n \times n}$. The constraint holds for *any* $a \in \mathbb{R}^n$, so $aa^t$ defines an infinite family of linear constraints. Thus in a sense $X$ satisfies infinite constraints, and the region of feasible solutions to an SDP is bounded by an infinite set of hyperplanes, as in Figure 2, for instance.

# 3 Vector Programming

Recall $X \in PSD_n$ iff $\exists v_1, \ldots, v_n \in \mathbb{R}^n$ s.t. $x_{ij} = \langle v_i, v_j \rangle$. That is, instead of finding the best matrix $X$ that satisfies a given set of constraints, we can think of finding an optimal set of vectors $V = \{v_1, \ldots, v_n\}$ with constraints on the vector inner products.

We write a *vector program* as

$$\begin{array}{cc} & \min \sum_{i,j} c_{ij} \langle v_i, v_j \rangle \qquad \text{s.t.} \\ \forall k = 1 \ldots m & \sum_{i,j} a_{ij}^{(k)} \langle v_i, v_j \rangle = b_k \\ & v_1, \ldots, v_n \in \mathbb{R}^n \end{array}$$

**Observation 3.1.** *This is identical to an SDP, just written in a different form. It is clear that a feasible solution here translates to a feasible SDP solution, by* $X = (x_{ij}) = (\langle v_i, v_j \rangle)$ *and vice versa, with* $C = (c_{ij})$ *and* $A_k = (a_{ij}^k)$*.*

**Definition 3.2.** A *metric* $d$ on $n$ points $\{1, \ldots, n\}$ is a function $d_{ij}$ s.t.

$$\begin{array}{ccll} d_{ij} & \geq & 0 & \text{(non-negative)} \\ d_{ij} & = & d_{ji} & \text{(symmetry)} \\ d_{ii} & = & 0 & \\ d_{ij} + d_{jk} & \geq & d_{ik} & \text{(triangle inequality)} \end{array}$$

Note that $d$ can be represented as an $n \times n$ symmetric matrix.

**Definition 3.3.** A metric $d$ on $n$ points is *Euclidean* if

$$\begin{array}{c} \exists p_1, \ldots, p_n \in \mathbb{R}^n \text{ s.t.} \\ \|p_i - p_j\|_2 = d_{ij} \end{array} \tag{2}$$

The set $\{p_1, \ldots, p_n\}$ is referred to as the Euclidean embedding of $d$.

**Observation 3.4.** *Not every metric is Euclidean.*

*Proof.* Consider Figure 3, for instance. Suppose $\{p_A, p_B, p_C, p_D\}$ is the Euclidean embedding. Since

$$\|p_A - p_B\|_2 + \|p_B - p_C\|_2 = d_{AB} + d_{BC} = d_{AC} = \|p_A - p_C\|_2$$

$p_A$, $p_B$ and $p_C$ must lie on a line with $p_B$ at the midpoint. Similarly, $p_A$, $p_B$ and $p_D$ must also lie on a line with $p_B$ at the midpoint. This collapses $p_C$ and $p_D$, implying $\|(p_c - p_d)\|_2 = 0$. However, this is inconsistent with $d_{CD} = 2$.

□

We relax (2) and define the *cost* of the embedding as the minimum $c$ such that

$$\forall i, j \in \{1, \ldots, n\}, d_{ij} \leq \|p_i - p_j\|_2 \leq c d_{ij}$$

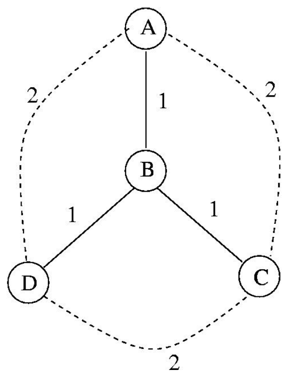


Figure 3: An example of a metric on four points that is not Euclidean.

The inequalities are preserved when squaring all terms,

$$d_{ij}^2 \leq \|p_i - p_j\|_2^2 \leq c^2 d_{ij}^2$$

Using the connection between the Euclidean norm and inner product,

$$\|p_i - p_j\|_2^2 = \langle p_i - p_j, p_i - p_j \rangle = \langle p_i, p_i \rangle + \langle p_j, p_j \rangle + 2\langle p_i, p_j \rangle$$

In other words, taking $\mathcal{C} = c^2$, we can formulate the problem of finding the minimum cost embedding as a VP instance.

$$\begin{gathered} \min \mathcal{C}, \text{ s.t. } \forall i, j \in \{1, \ldots, n\}, \\ d_{ij}^2 \leq \langle p_i, p_i \rangle + \langle p_j, p_j \rangle - 2\langle p_i, p_j \rangle \leq \mathcal{C} d_{ij}^2 \end{gathered}$$

**Observation 3.5.** *We can't hope to add a dimension constraint to the above, for example to ask for the vectors to be of dimension 5. Vector programming with restrictions on dimension is NP-hard.*

**Example 3.6.** Consider the Vertex cover problem. Given a graph $G = (V, E)$, find the smallest set of vertices $S \in V$ such that for all edges $\{i, j\} \in E$ either vertex $i \in S$ or $v_j \in S$ (or both).

Suppose we have a VP in which the vectors are restricted to dimension one. We have variables $x_i$ for vertices $i \in V$, and $x_0$ (an indicator value).

$$\begin{gathered} \min \sum_i \langle x_0, x_i \rangle \\ \|x_i\|_2 = 1 \\ \forall i, j \in E, \langle x_i, x_0 \rangle + \langle x_j, x_0 \rangle \geq 0 \\ x_i \in \mathbb{R}^1 \end{gathered}$$

**Claim 3.7.** *A solution to the above translates to an exact solution to vertex cover, where $S = \{v_i \mid x_i = x_0\}$ and $\sum_i \langle x_0, x_i \rangle = 2|S| - n$.*

## 4 Duality in SDP

As in LP, we should think of the dual as a linear method that bounds the optimimum. To bound $C \bullet X$ from below, we want to:

(i) Find coefficients $y_i$ for each of the $m$ constraints

(ii) Guarantee that $\forall X \succeq 0$, it holds that

$$\sum_{i=1}^{m} y_i A_i \bullet X \leq C \bullet X \tag{3}$$

Having done so, since $\sum y_i A_i = \sum y_i b_i$, we get a lower bound on the objective of the primal: $C \bullet X \geq \langle y, b \rangle$. What restrictions must we put on $y$ so that (3) holds? Rewrite (3) as

$$\forall X \succeq 0, (C - \sum_{i=1}^{m} y_i A_i) \bullet X \geq 0$$

In particular, for $x \in \mathbb{R}^n$, we have that $xx^t \succeq 0$ and so $C - \sum y_i A_i$ must satisfy

$$\forall x \in \mathbb{R}^n, (C - \sum y_i A_i) \bullet (xx^t) = x^t (C - \sum y_i A_i) x \geq 0$$

Therefore, to satisfy (3), $(C - \sum y_i A_i)$ must be $PSD$.
On the other hand, if $X \succeq 0$ then

$$X = V^t V = \sum_i w_i w_i^t, \text{ where } V = \begin{pmatrix} -w_1- \\ \vdots \\ -w_n- \end{pmatrix}$$

That is, every $X \succeq 0$ can be represented as the non-negative sum of outer-products and so, if $(C - \sum_{i=1}^{m} y_i A_i)$ satisfies (3) for every $xx^t$, it satisfies it for every $X \succeq 0$.

**Definition 4.1.** Let $K$ be a cone. The *dual cone* $K^*$ is defined

$$K^* = \{ y \mid \langle x, y \rangle \geq 0, \forall x \in K \}$$

In formulating the dual of LP, we use $((\mathbb{R}^+)^n)^* = (\mathbb{R}^+)^n$. We now need to understand the dual of $PSD_n$

**Claim 4.2.** $PSD_n^* = PSD_n$

We write the dual as

$$\begin{array}{cc} \underline{\text{Primal}} & \underline{\text{Dual}} \\ \min C \cdot X & \max \langle y, b \rangle \\ i = 1 \ldots m, A_i \bullet X = b_i & \sum y_i A_i - C \succeq 0 \\ X \succeq 0 & y \lesseqgtr 0 \end{array}$$

Note that the constraint $\sum y_i A_i - C \succeq 0$ is equivalent to $\sum y_i A_i \preceq C$.

# 5 Differences between LP and SDP

Many of the “nice” LP properties do not carry over for SDP.

- The optimum cannot always be attained, even if the problem is bounded
- Even if the optimum can be attained, it may be irrational, and so can’t be solved exactly on a finite machine
- There is no analogue in SDP for LP’s Basic Feasible Solutions (BFS)
- There *can* be a duality gap. See theorem 5.1, below. Strong duality does not neccessarily hold.
- Implied by the above, there is no natural bound on the size of the solution in terms of input size.

**Theorem 5.1.** *If there is a feasible solution $X \succ 0$ (all eigenvalues strictly positive) with $C - \sum y_i A_i \succ 0$ then both primal and dual attain optimum and they are the same.*