1 Recap: Interpretation of SDPs

Last time, we saw two important interpretations of

\[ X \succeq 0 \]

i.e. the statement that \( X \) is PSD:

1. \( X \) is an inner product matrix, i.e. \( X_{ij} = v_i^T v_j \) for vectors \( v \).
2. \( X \) is a density matrix corresponding to a distribution over vectors, e.g.

\[ X = E[zz^T] \]

where \( z \ D \) and \( D \) is a distribution of vectors on the \( n-1 \) sphere.

We elaborate on the second interpretation. First, we can construct a density matrix for any distribution \( D \) of vectors \( x \) over the \( n-1 \) sphere. Let \( p_x \) \( D \) be the probability of vector \( x \). Then we have

\[ X = E_D[xx^T] = \sum_x p_xx x^T . \]

\( xx^T \) is PSD and \( p_x \) is positive, so \( E_D[xx^T] \) is also PSD. Additionally, we note that the trace of \( X \) must be 1. We have

\[ Tr(X) = \sum_x p_x Tr(xx^T) = \sum_x p_x ||x|| . \]

Since \( x \) is on the unit sphere, \( ||x|| = 1 \). Thus,

\[ Tr(X) = \sum_x p_x = 1 \]
because $p_x$ is a probability distribution.

Second, any PSD matrix $X \succeq 0$ with $Tr(X) = 1$ may be interpreted as such an expectation. Consider the eigendecomposition

$$X = \sum_i \lambda_i v_i v_i^T.$$ 

Since $Tr(X) = \sum \lambda_i = 1$ and $X$ is PSD (so $\lambda_i \geq 0$), it follows that the $\lambda_i$ represent a valid probability distribution over the vectors $v_i$.

Finally, we consider the Frobenius product in this context. What is the meaning of $A \bullet X$ when $X$ is PSD? Interpreting $X$ as a distribution over vectors we have

$$A \bullet X = \sum_x p_x(A \bullet xx^T).$$

Using the definition of the Frobenius product, this is equivalent to $\sum_x p_x(x^T Ax)$ which gives

$$A \bullet X = \sum_x p_x(x^T Ax) = E_D[x^T Ax].$$

## 2 Interpretations of the MAXCUT SDP

The MAXCUT problem: given a graph $G = (V, E)$, find a set of vertices $S \subset V$ such that the number of edges from $S$ to $\bar{S} = V \setminus S$ is maximized, i.e. we maximize $|E(S, \bar{S})|$. The problem is NP-hard; however, a trivial randomized approximation algorithm achieves an approximation ratio of 0.5.

**An Integer Program**

We can write the MAXCUT problem as the following integer program:

$$\max \frac{1}{4} \sum_{(i,j) \in E} (x_i - x_j)^2$$

$$x_i \in \{-1, 1\}.$$ 

Our next task is to relax the integer program.
Relaxation 1

As a first pass, we try the following vector formulation

$$\max \frac{1}{4} \sum_{(i,j) \in E} ||v_i - v_j||^2$$

$$\sum_{i \in V} d_i ||v_i||^2 = 2|E|$$

The matrix form of this SDP is

$$\max \frac{1}{4} L \cdot X$$

$$D \cdot X = 2|E|$$

where $L = D \cdot A$ is the Lagrangian of the graph, $D$ is the degree matrix, and $A$ is the adjacency matrix. This may be rephrased as a generalized eigenvalue problem

$$\max \lambda$$

$$Ax = \lambda Dx$$

or equivalently

$$\max \lambda$$

$$\frac{1}{4} L - \lambda D \preceq 0$$

Trevisan (roughly) used this SDP to get a 0.531 approximation.

Relaxation 2

The more well-known relaxation due to Goemans and Williamson merely replaces the vector normalization constraints. The vector form is

$$\max \frac{1}{4} \sum_{(i,j) \in E} ||v_i - v_j||^2$$

$$\forall i \in V: ||v_i||^2 = 1$$

In essence, all vectors $v_i$ must lie on the unit sphere. The matrix form is

$$\max \frac{1}{4} L \cdot X$$

$$\forall i : X_{ii} = 1$$
\[ X \succeq 0 \]

Considering \( X \) to be a distribution of vectors gives the interpretation

\[
\max \frac{1}{4} E_D[x^T L x]
\]

\[ E[x_i^2] = 1 \]

where the maximization problem is over distributions \( D \).

The dual also admits an interesting interpretation. The dual SDP is

\[
\min \sum_{i \in V} \beta_i
\]

\[
\frac{1}{4} L - \sum_{i \in V} \beta_i (e_i e_i^T) \preceq 0 .
\]

The matrix \( e_i e_i^T \) corresponds to the graph of a star centered at \( i \). It is not immediately clear why the star should appear here.

**Rounding Relaxation 2**

We think about rounding as an embedding problem. As noted earlier, the vectors in the SDP are embedded on the unit sphere. They are rounded by choosing a random hyperplane through the center of the sphere.

The probability that two vectors lie on opposite sides of the hyperplane is \( \frac{\theta}{\pi} \) where \( \theta \) is the angle between the two vectors, i.e. \( \theta = \frac{\arccos(v_i^T v_j)}{\pi} \) for vectors \( v_i \) and \( v_j \). Thus, the expected number of edges cut is

\[
E[\# \text{ of edges cut}] = \sum_{(i,j) \in E} \frac{\arccos(v_i^T v_j)}{\pi} .
\]

The approximation ratio of this algorithm can be established by relating \( \arccos(v_i^T v_j) \) to \( v_i^T v_j \).