Convex and Nonsmooth Optimization
HW4: Mostly about Semidefinite Programming

Michael Overton

Spring 2020

1. (20 pts) BV Ex 5.13

2. (40 pts) Consider the primal SDP

$$\begin{aligned}&\min \langle C, X\rangle\\ \text{subject to }&\langle A_i, X\rangle = b_i, \quad i = 1, 2\\ &X \succeq 0\end{aligned}$$

with

$$C = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, A_1 = \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}, A_2 = \begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}, b_1 = 1, b_2 = 0$$

(a) Does the Slater condition hold for this primal SDP, i.e., does there exist a strictly feasible $\tilde{X}$?

Let $\tilde{X} = \begin{bmatrix}a & b\\ c & d\end{bmatrix}$ For a feasible $\tilde{X}$ to exists, the equality constraints need to be satisfied:

$$\left\langle A_1, \tilde{X}\right\rangle = b_1 \Rightarrow \operatorname{tr}\left(\begin{bmatrix}a & b\\ 0 & 0\end{bmatrix}\right) = 1 \Rightarrow a = 1$$

$$\left\langle A_2, \tilde{X}\right\rangle = b_2 \Rightarrow \operatorname{tr}\left(\begin{bmatrix}0 & 0\\ c & d\end{bmatrix}\right) = 0 \Rightarrow d = 0$$

We are then looking for $\tilde{X} = \begin{bmatrix}1 & b\\ c & 0\end{bmatrix}$ s.t $-\tilde{X} \prec 0$, equivalently $\tilde{X} = \begin{bmatrix}1 & b\\ b & 0\end{bmatrix}$ s.t $-\tilde{X} \prec 0$, or $-\tilde{X} = \begin{bmatrix}-1 & -b\\ -b & 0\end{bmatrix} \prec 0$, which would imply that this matrix has all its eigenvalues negative. Its characteristic equation is: $x^2 + x - b^2 = 0$ the discriminant is $\Delta = 1 + 4b^2 \geq 0$, thus there is one or two roots satisfying the conditions $x_1 + x_2 = -1, x_1\ x_2 = -b^2 < 0$, the last inequality is impossible if the roots have to be negative. The slater condition does not hold for this primal SDP.

(b) What is the optimal value of the primal SDP? Is it attained, and if so, by what X?
When the equality constraints are satisfied, the optimization problem can be expressed as

$$\begin{aligned}&\min \langle C, X\rangle\\ &\text{subject to } X \succeq 0,\ \text{where } X = \begin{bmatrix}1 & b\\ b & 0\end{bmatrix}\end{aligned}$$

or equivalently

$$\min \operatorname{tr}\left(\begin{bmatrix} b & 1 \\ 0 & b \end{bmatrix}\right) = \min b$$
$$\text{subject to } X \succeq 0, \text{ where } X = \begin{bmatrix} 1 & b \\ b & 0 \end{bmatrix}$$

X has for polynomial characteristic: $P(\lambda) = \lambda^2 - \lambda - b^2$. X is positive semi definite if and only if all of its eigenvalues are non-negative. This quadratic has zero, one or two roots. For X to have non negative eigenvalues then the equation has to satisfy the constraints: $\lambda_1 + \lambda_2 = 1, \lambda_1\ \lambda_2 = -b^2 \geq 0$ which implies that one of the eigenvalue is 0 and the other 1 and $b = 0$. Thus the primal value is $p^* = 0$ and its attained by $X = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$.

(c) Write down the dual SDP.

The Lagrangian is: $L(X, \Lambda, \nu) = \langle C, X\rangle - \langle \Lambda, X\rangle + \sum_{i=1,2} \nu_i(\langle A_i, X\rangle - b_i)$ The Lagrangian dual function is $g(\Lambda, \nu) = \inf_X(\langle C - \Lambda, X\rangle + \sum_{i=1,2} \nu_i \langle A_i, X\rangle) - \nu^T b$. So we can write the dual problem as

$$g(\Lambda, \nu) = \begin{cases} -\nu^T b & \text{if } \sum_{i=1,2} \nu_i A_i = \Lambda - C \\ -\infty & \text{otherwise} \end{cases}$$

Let $y = -\nu$ that is,

$$\sup_{\Lambda, y} b^T y$$
$$\text{subject to } \sum_{i=1,2} y_i A_i + \Lambda = C$$

or equivalently

$$\sup \Lambda, y b^T y$$
$$\text{subject to } (C - \sum_{i=1,2} y_i A_i) \succeq 0$$

With given C, A_1, A_2, b_1, b_2, let $y = [y_1\ y_2]^T$, the dual SDP is

$$\sup y_1$$
$$\text{subject to } \begin{bmatrix} -y_1 & 1 \\ 1 & -y_2 \end{bmatrix} \succeq 0$$

(d) Does the Slater condition hold for the dual SDP, i.e., does there exist a strictly feasible dual variable $\tilde{y}$?

We are looking for a dual variable $\tilde{y}$ which satisfies the strict inequality: $\begin{bmatrix} -y_1 & 1 \\ 1 & -y_2 \end{bmatrix} \succ 0$. The polynomial characteristic of this matrix is: $P(x) = x^2 + (y_1 + y_2)x + y_1 y_2 - 1$, $P(x) = 0 \Leftrightarrow x^2 + (y_1 + y_2)x + y_1 y_2 - 1 = 0$. Eigenvalues of this matrix verify the equations:

$$x_1 + x_2 = -(y_1 + y_2)$$
$$x_1\ x_2 = y_1\ y_2 - 1$$

If the eigenvalues are positive then $x_1 \ x_2 > 0 \Rightarrow (y_1 + y_2) < 0 \ , y_1 \ y_2 > 1$. We can choose $y_1 = y_2 = -2$ and $\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \succ 0$ (the eigenvalues are 1 and 3). A strictly feasible dual variable is $\tilde{y} = [-2 - 2]^T$.

(e) What is the optimal value of the dual SDP? Is it attained, and if so, by what dual variable y? Taking $y_2 \to -\infty, y_1 = \frac{1}{y_2}$ verifies that $\begin{bmatrix} -y_1 & 1 \\ 1 & -y_2 \end{bmatrix} \succeq 0$ since the eigenvalues are 0 and $-(\frac{1}{y_2} + y_2) > 0$. The optimal value is $d^* = 0$ and it is attained by $y = \sup_{z>0}[\frac{1}{z} \ z]^T$.

(f) Does strong duality hold?
Since $d^* = p^*$ the strong duality holds.

(g) What can you say in general about strong duality if the Slater condition holds for at least one of a primal-dual pair of SDPs? If the problem is complex and Slater's condition holds, it implies strong duality. If Slater's condition holds for the dual strong, duality can or cannot hold.

3. (40 pts) Exercise 5.39 on p. 285–286 in BV (see also p. 219–220). Assume that the matrix W is componentwise nonnegative, so that $W_{ij} = W_{ji}$ can be interpreted as a nonnegative weight on the edge joining vertex i to vertex j. As well as answering the questions in the exercise, also do the following.

(a) Two-way partitioning problem in matrix form. Show that the two-way partitioning problem can be cast as

$$\begin{aligned} &\text{minimize } \operatorname{tr}(\mathbf{W}\mathbf{X}) \\ &\text{subject to } \mathbf{X} \succeq 0, \ \textbf{rank} \ = 1 \\ &X_{ii} = 1, i = 1, \ldots, n \end{aligned}$$

First, let $\mathbf{X} = \boldsymbol{x}\boldsymbol{x}^T$, $\mathbf{X}$ has rank 1 since it can be written as: $\sum_{i=1,n} x_i \boldsymbol{x}, i = 1, \ldots, n$. We have $\boldsymbol{x}^T \mathbf{W} \boldsymbol{x} = \sum_{i,j=1,n} x_i x_j W_{i,j} = \sum_{i,j} W_{i,j=1,n} x_i x_j = \operatorname{tr}\left(W \boldsymbol{x}\boldsymbol{x}^T\right)$ where $X_{ii} = x_i^2, i = 1, \ldots, n$. So the initial problem:

$$\begin{aligned} &\text{minimize } \boldsymbol{x}^T \mathbf{W} \boldsymbol{x} \\ &\text{subject to } x_i^2 = 1 \ i = 1, \ldots, n \end{aligned}$$

can be written as

$$\begin{aligned} &\text{minimize } \operatorname{tr}(\mathbf{W}\mathbf{X}) \\ &\text{subject to } \mathbf{X} \succeq 0, \ \textbf{rank} \ = 1 \\ &X_{ii} = 1, i = 1, \ldots, n \end{aligned}$$

(b) SDP relaxation of two-way partitioning problem. Using the formulation in part (a), we can form the relaxation

$$\begin{aligned} &\text{minimize } \operatorname{tr}(\mathbf{W}\mathbf{X}) \\ &\text{subject to } \mathbf{X} \succeq 0, \ \textbf{rank} \ = 1 \\ &X_{ii} = 1, \ i = 1, \ldots, n \end{aligned}$$

with variable $\mathbf{X} \in \mathbf{S}^n$ This problem is an SDP, and therefore can be solved efficiently. Explain why its optimal value gives a lower bound on the optimal value of the two-way partitioning problem (5.113). What can you say if an optimal point $\mathbf{X}^*$ for this SDP has rank one?

The Lagrangian for the SDP relaxation of two-way partitioning problem, is :

$$
\begin{aligned}
L(\mathbf{X}, \mathbf{\Lambda}, \boldsymbol{\nu}) &= \langle \mathbf{W}, \mathbf{X} \rangle - \langle \mathbf{\Lambda}, \mathbf{X} \rangle + \sum_{i=1}^{n} \nu_i (X_{ii}^2 - 1) \\
&= \langle \mathbf{W} - \mathbf{\Lambda}, \mathbf{X} \rangle + \langle \mathbf{diag}(\nu), \mathbf{X} \rangle - \mathbf{1}^T \boldsymbol{\nu} \\
&= \langle \mathbf{W} - \mathbf{\Lambda} + \mathbf{diag}(\nu), \mathbf{X} \rangle - \mathbf{1}^T \boldsymbol{\nu}
\end{aligned}
$$

We obtain the Lagrange dual function by minimizing over $\mathbf{X}, \mathbf{X} \in \mathbf{S}^n$:

$$
g(\boldsymbol{\nu}) = \inf_{\mathbf{X}, \mathbf{X} \in \mathbf{S}^n} (\langle \mathbf{W} - \mathbf{\Lambda} + \mathbf{diag}(\nu), \mathbf{X} \rangle) - \mathbf{1}^T \boldsymbol{\nu} = \begin{cases} -\mathbf{1}^T \boldsymbol{\nu} & \text{if } \mathbf{\Lambda} - \mathbf{W} = \mathbf{diag}(\nu) \Leftrightarrow \mathbf{W} + \mathbf{diag}(\nu) \succeq 0 \\ -\infty & \text{otherwise} \end{cases}
$$

The constraints of the original problem imply the constraints on problem found in part (a). In the original problem we minimize over $\mathbb{R}^n$ and in the later we minimize over $\mathbf{S}^n$, a larger set so the optimal value for the SDP relaxation will be lower bound to the original problem.

(c) We now have two SDPs that give a lower bound on the optimal value of the two-way partitioning problem (5.113): the SDP relaxation (5.115) found in part (b), and the Lagrange dual of the two-way partitioning problem, given in (5.114). What is the relation between the two SDPs? What can you say about the lower bounds found by them? Hint: Relate the two SDPs via duality.

We have show in part (b) that the dual problem of the SDP relaxation (5.115) and the dual problem of the two-way partitioning problem, given in (5.114) are the same hence the lower bounds found by them are the same.

First, here is some useful information.

- The easiest way to set up an SDP in CVX is to use "cvx_begin sdp" instead of "cvx_begin". Then, all matrix inequalities before the next "cvx_end" will be interpreted as semidefinite inequalities. Be sure to declare any symmetric matrix variables as symmetric, like this: "variable X(n,n) symmetric". See here for more details.
- If you declare a variable as "dual variable Z" and then put ": Z" after an equality or inequality constraint, you will have access to the computed optimal dual variable for that constraint. If you want more than one dual variable, use "dual variables Z y" (no comma).
- In MATLAB, and hence also CVX, if X is a matrix, diag(X) is a vector, and if x is a vector, diag(x) is a matrix. Type "help diag" for more.
- Because the rank of a matrix is a discontinuous function, "rank(X)" is not a reliable way to find the approximate rank of a matrix, especially one that has been computed with CVX. Instead, compute the eigenvalues with "eig" (assuming X is symmetric) and estimate the rank from the eigenvalues.

(a) Using W from data set 1 and data set 2, solve the SDP given in BV (5.114) with CVX.

Using W from these two datasets and solving the SDP given in BV (5.114) we found for the first data set an optimal value of -15 and the second data set an optimal value of -130.55.

Please refer at the end of the documentation to the relevant code.

(b) Also, solve the SDP given in BV (5.115) by CVX and compare its optimal value with the one for (5.114). For the smaller data set 1, compare the computed optimal dual variables from (5.115) with the computed optimal primal variables from (5.114) and vice versa. What are the approximate ranks of the computed optimal primal and dual matrices? Do the matrices satisfy approximate complementarity, e.g. is the matrix product $X * Z$ approximately zero?

Using the primal problem expressed in (5.115) we obtain the same optimal values obtained using (5.114). The computed dual variables from 5.115 are the same as the computed primal variables from 5.114, and vice-versa. The rank of the computed optimal optimal primal and dual matrices from the primal problem (5.115) are the same and equal to 10. These two matrices satisfy approximate complementarity.

Please refer at the end of the documentation to the relevant code. Output from the Matlab script:
Found for data set 1, Lagrange dual problem, optimal value:-15.00
Found for data set 1, SDP relaxation of two-way partitioning problem,optimal value:-15.00
Computed dual variables from 5.115 same as computed primal variables from 5.114
Computed primal variables from 5.115 same as computed dual variables from 5.114
Rank(optimal primal matrix): 10, Rank(optimal dual matrix): 10
Computed optimal and dual matrices are complementary, tol:0.00010
Found for data set 2, Lagrange dual problem, optimal value:-130.55
Found for data set 2, SDP relaxation of two-way partitioning problem,optimal value:-130.55

(c) Here is another way to motivate the SDP relaxation (5.115). Instead of insisting that the variables x_i in (5.113) have the values ± 1, replace each scalar x_i by a vector $v_i \in \mathbb{R}^n$ with $\|v_i\|_2 = 1$, and then write $V = [v_1, \dots, v_n]$ and $X = V^T V$. Does such an X satisfy the constraints in (5.115)? This is the motivation used in Goemans and Williamson's celebrated 1994 paper and this leads to a simple randomized procedure for assigning the vertices to the two sets: see equations (1)–(3) on p. 1120 of their paper. Solving the SDP gives you $X \succeq 0$ and then you need V such that $X = V^T V$. Is such a V unique? If not, what is a convenient choice? Show that it gives you V whose columns have norm one as required. Would this work if X is exactly low rank, or low rank to machine precision, instead of only approximately low rank? Why or why not?

Using this V, carry out the assignment algorithm on p. 1120 for the data sets 1 and 2 using r with $r_i = 1/\sqrt{n}$ (instead of a random vector) and print the resulting partitioning of the vertices and the corresponding cut value. How does it compare to the optimal value of the SDP?

The problem is to find an optimal partition of n elements, in Goemans and Williamson's celebrated 1994 paper, given non negative weights, and the symmetric matrix of these weights W, we want to solve the following integer quadratic program:

$$\begin{aligned} &\text{maximize } \frac{1}{2} \sum_{i<j} w_{ij}(1 - x_i x_j) \\ &\text{subject to } x_i \in \{-1, 1\}, i = 1, \dots, n \end{aligned}$$

We can replace these products $1 - x_i x_j$ by the dot product of n vectors $\boldsymbol{v_i}, i = 1, \dots, n, \in \mathbb{R}^n$ with the constraint that their l_2 norm is 1.

Or equivalently we want to find the vector $\boldsymbol{x} \in \mathbb{R}^n$ which minimizes $\sum_{ij} x_i W_{ij} x_j = \boldsymbol{x}^T \mathbf{W} \boldsymbol{x}$ subject to $x_i^2 = 1, i = 1, \dots, n$. In BV exercise 5.39, we showed that, this last problem lead to the

SPD relaxation problem (5.115):

$$\begin{aligned}&\text{minimize } \operatorname{tr}(\mathbf{W}\mathbf{X})\\ &\text{subject to } \mathbf{X} \succeq 0\\ &X_{ii} = 1,\ i = 1, \dots, n\end{aligned}$$

with variable $\mathbf{X} = \boldsymbol{x}\boldsymbol{x}^T, \mathbf{X} \in \mathbf{S}^n$.

$$\mathbf{V}^T\mathbf{V} = \begin{bmatrix} \text{———} & v_1 & \text{———} \\ \text{———} & \vdots & \text{———} \\ \text{———} & v_n & \text{———} \end{bmatrix} \begin{bmatrix} v_1 & | & \dots & | & v_n \end{bmatrix} = \begin{bmatrix} v_1^T v_1 & \dots & v_n^T v_1 \\ \vdots & \ddots & \vdots \\ v_1^T v_n & \dots & v_n^T v_n \end{bmatrix}$$

$\mathbf{X} = \mathbf{V}^T\mathbf{V}$ satisfies the constraints $\mathbf{X} \succeq 0$ (any matrix $\mathbf{A}^T\ \mathbf{A}$ or $\mathbf{A}^T\ \mathbf{A}$ is always positive definite). In addition, $\mathbf{X}_{ii} = \|v_i\|_2^2 = 1,\ i = 1, \dots, n$ then $\mathbf{X}$ satisfies the constraints in (5.115). We can choose these vectors $\boldsymbol{v}_i$ to belong to the n-dimensional unit sphere S_n hence they are not unique. Low rank to machine precision for $\mathbf{X}$ will provide a suboptimal solution since the optimal solution is the solution with exact low rank.

Implementing the assignment algorithm on p. 1120 for the data sets 1 and 2 using r with $r_i = 1/\sqrt{n}$ we obtain the results herein:

Max. cut value data set 1: 58 Partition for data set 1: 1 0 1 1 0 1 1 1 1 1

Max. cut value data set 2:697.28 Partition for data set 2 0 0 0 1 0 0 1 1 1 0 1 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Please refer at the end of the documentation to the relevant code.

(d) Explicitly solve (5.113) for data set 1 only. This problem is NP-hard so you will have to write a brute force method to solve it: there is no way to do it efficiently, but it should run fast enough on the smaller data set 1. According to Goemans and Williamson, the optimal value in their SDP relaxation (which CVX computes in polynomial time up to a given accuracy) should be within a factor of ≈ 0.878 of the optimal value of the max cut problem. Is it? If not, perhaps there are some issues of scaling or constants that need working through.

An additional note of interest, not part of the homework: Hastad showed that there is no polynomial time algorithm to improve this max-cut guaranteed approximation factor from 0.878 to $16/17 \approx 0.941$ (assuming P $\neq$ NP), and Courant's Subhash Khot and his collaborators showed in this 2005 paper that if Subhash's "unique games conjecture" is true, then SDP is optimal for max-cut: one cannot get a better approximation guarantee than ≈ 0.878 in polynomial time unless P=NP. This would mean that SDP is somehow a very fundamental notion. Amazing!

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%% Problem 3 Question a %%%%%%%%%%%%%%%%%%%%%%%

% Problem 3 - Question (a)
% Solve the SDP given in BV (5.114) using data set 1 and data set 2


W = load("hw4data1.mat").W;
solve_dual(W)
fprintf("Found for data set 1, optimal value:%4.2f\n",  cvx_optval)
```

```matlab
W = load("hw4data2.mat").W;
solve_dual(W)
fprintf("Found for data set 2, optimal value:%4.2f\n",  cvx_optval)


function solve_dual(W)

[n,~]=size(W);

cvx_begin sdp
    variable nu(n)
    maximize (-sum(nu, 1))
    W + diag(nu) == semidefinite(n)
cvx_end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%% Problem 3 Question b %%%%%%%%%%%%%%%%%%%%%%%

% Problem 3 - Question (b) -
% Solve the SDP given in BV (5.114) using data set 1 and data set 2
% In addition solve BV (5.115) using data set 1:
% 1- Compare computed optimal and dual variables from (5.115)
%    with computed optimal primal variables from (5.114) and vice-versa
% 2- Compute approximate ranks of the computed optimal primal and dual
%    matrices
% 3- Test for complimentarity of these matrices.


clear
cvx_quiet true

W = load("hw4data1.mat").W;
[opt_val, X1, nu1] = solve_dual(W);
fprintf("Found for data set 1,
 Lagrange dual problem, optimal value:%4.2f\n",  opt_val)

[opt_val, X2, nu2, lambda] = solve_sdp_relaxation(W);
fprintf("Found for data set 1,
SDP relaxation of two-way partitioning problem,optimal value:%4.2f\n",  opt_val)

if is_equal(abs(nu1), abs(nu2))
    disp("Computed dual variables from 5.115
    same as computed primal variables from 5.114")
else
    disp("Computed dual variables from 5.115
    different from the computed primal variables from 5.114")
end
```

```
if is_equal(X1, X2)
    disp("Computed primal variables from 5.115
    same as computed dual variables from 5.114")
else
    disp("Computed primal variables from 5.115
    different than computed dual variables from 5.114")
end

fprintf("Rank(optimal primal matrix): %d, Rank(optimal dual matrix): %d\n",...
    compute_rank(X2) , compute_rank(lambda))

tol = 1e-4;
if are_complementary(X2, lambda, tol)
    fprintf("Computed optimal and dual matrices are complementary,
    tol:%2.5f\n", tol)
else
    fprintf("Computed optimal and dual matrices are not complementary,
    tol:%5.2f\n", tol)
end

clear
W = load("hw4data2.mat").W;
[opt_val, ~, ~] = solve_dual(W);
fprintf("Found for data set 2,
Lagrange dual problem, optimal value:%4.2f\n",  opt_val)
[opt_val, ~, ~, ~] = solve_sdp_relaxation(W);
fprintf("Found for data set 2,
SDP relaxation of two-way partitioning problem,optimal value:%4.2f\n",  opt_val)

function [opt_val, X, nu] = solve_dual(W)

[n,~]=size(W);

cvx_begin sdp
    variable nu(n)
    dual variable X
    maximize (-sum(nu, 1))
    X: W + diag(nu) == semidefinite(n);
cvx_end
opt_val = cvx_optval;
end

function [opt_val, X, nu, lambda]  = solve_sdp_relaxation(W)

[n,~]=size(W);

cvx_begin sdp
```

```
    variable X(n, n) symmetric
    dual variables nu lambda
    minimize trace(W * X)
    nu: diag(X) == 1;
    lambda: X == semidefinite(n);
cvx_end
opt_val = cvx_optval;

end

function t = is_equal(A, B)
% We are expecting that A and B have the same dimensions.
% We could eventually perform a check for this assumption.

    [m, n] = size(A);
    if (sum(A == B, "all") == m * n)
        t = true;
    else
        t = false;
    end
end

function rk = compute_rank(A)
% Assume A is symmetric
D = eig(A);
[n, ~] = size(A);
k = 0;
for i=1:n
    if D(i,1) == 0
        k = k + 1;
    end
end
rk = n - k;
end

function t = are_complementary(A, B, tol)
% Testing if A * B ~ 0
    t = isempty(find(A* B > tol));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%% Problem 3 Question c %%%%%%%%%%%%%%%%%%%%%%
% Implementation of MAXCUT algorithm
% Using data set 1 and data set 2 to determine
% maxcut and the corresponding partition.

clear
cvx_quiet true
```

```matlab
W = load("hw4data1.mat").W;
[n, ~] = size(W);
[X, cut_value] = maxcut(n, W);
V = chol(X);
fprintf("Max. cut value data set 1:%5.2f\n", cut_value)
[partition] = get_partition(V, n);
fprintf("Partition for data set 1\n")
disp(partition')

clear
W = load("hw4data2.mat").W;
[n, ~] = size(W);
[X, cut_value] = maxcut(n, W);
V = chol(X);
fprintf("Max. cut value data set 2:%5.2f\n", cut_value)
[partition] = get_partition(V, n);
fprintf("Partition for data set 2\n")
disp(partition')

function [X, cut_value] = maxcut(n, W)
cvx_begin sdp
    variable X(n,n) symmetric
    maximize 0.5 .* sum(sum(W .* (ones(n,n) - X)))
    diag(X) == 1;
    X >= 0;
cvx_end
cut_value = cvx_optval;

end

function [partition] = get_partition(V, n)
partition = zeros(n, 1);
r = (1/sqrt(n)) .* ones(n, 1);

for i=1:n
    if dot(V(i,:), r) >= 0
        partition(i) = 1;
    else
        partition(i) = 0;
    end
end
end
```