EE 150 - Applications of Convex Optimization in Signal Processing and Communications
Dr. Andre Tkacenko, JPL
Third Term 2011-2012

Homework Set #8 - Solutions

1. For all parts to this problem, we will use the fact that $H_R(f)$ can be expressed as

$$H_R(f) = \mathbf{c}(f)^T \mathbf{b}\,, \text{ where } [\mathbf{c}(f)]_k = \cos(2\pi\,(k-1)\,f)\ ,\ k = 1, \ldots, M+1\,.$$

In addition, we will define the following functions for sake of convenience.

$$\begin{aligned}
g_1(\mathbf{b}) &\triangleq \sup_{0 \le f \le f_P} H_R(f) = \sup_{0 \le f \le f_P} \mathbf{c}(f)^T \mathbf{b}\,,\\
g_2(\mathbf{b}) &\triangleq \inf_{0 \le f \le f_P} H_R(f) = \inf_{0 \le f \le f_P} \mathbf{c}(f)^T \mathbf{b}\,,\\
g_3(\mathbf{b}) &\triangleq \sup_{f_S \le f \le 1/2} H_R(f) = \sup_{f_S \le f \le 1/2} \mathbf{c}(f)^T \mathbf{b}\,,\\
g_4(\mathbf{b}) &\triangleq \inf_{f_S \le f \le 1/2} H_R(f) = \inf_{f_S \le f \le 1/2} \mathbf{c}(f)^T \mathbf{b}\,.
\end{aligned}$$

Note that as g_1 and g_3 are each the pointwise supremum of an affine (actually linear) function of $\mathbf{b}$, they are each convex. Similarly, as g_2 and g_4 are each the pointwise infimum of an affine (or rather linear) function of $\mathbf{b}$, they are each concave.

(a) The optimization problem here is the following.

$$\begin{array}{ll}
\text{minimize} & \delta_S \\
\text{subject to} & g_1(\mathbf{b}) \le 1 + \delta_P \\
& g_2(\mathbf{b}) \ge 1 - \delta_P \\
& g_3(\mathbf{b}) \le \delta_S \\
& g_4(\mathbf{b}) \ge -\delta_S
\end{array}\ ,$$

with variables $\delta_S \in \mathbb{R}$ and $\mathbf{b} \in \mathbb{R}^{M+1}$. This is identical to the following problem.

$$\begin{array}{ll}
\text{minimize} & \delta_S \\
\text{subject to} & g_1(\mathbf{b}) - (1 + \delta_P) \le 0 \\
& -g_2(\mathbf{b}) + (1 - \delta_P) \le 0 \\
& g_3(\mathbf{b}) - \delta_S \le 0 \\
& -g_4(\mathbf{b}) - \delta_S \le 0
\end{array}\ ,$$

As the objective δ_S is clearly convex, g_1 and g_3 are convex, and g_2 and g_4 are concave, this problem is a convex optimization problem.

(b) Define the function $g_5(\mathbf{b})$ as follows.

$$\begin{aligned}
g_5(\mathbf{b}) &\triangleq \inf\left\{\phi : -\delta_S \le H_R(f) \le \delta_S \text{ for } \phi \le f \le 1/2\right\}\ ,\\
&= \inf\left\{\phi : -\delta_S \le \mathbf{c}(f)^T \mathbf{b} \le \delta_S \text{ for } \phi \le f \le 1/2\right\}\ .
\end{aligned}$$

Note that $g_5(\mathbf{b})$ is quasiconvex as a function of $\mathbf{b}$. To see this, note that the ϕ_0-sublevel set $\mathcal{C}_{\phi_0}$ is given by

$$\mathcal{C}_{\phi_0} = \{\mathbf{b} : g_5(\mathbf{b}) \le \phi_0\} = \left\{\mathbf{b} : -\delta_S \le \mathbf{c}(f)^T \mathbf{b} \le \delta_S \text{ for } \phi_0 \le f \le 1/2\right\}\ ,$$

which is the intersection of an infinite number of halfspaces. As such, $\mathcal{C}_{\phi_0}$ is convex, and so $g_5(\mathbf{b})$ is quasiconvex.

With the function g_5 defined as such, the optimization problem becomes the following.

$$\begin{array}{ll} \text{minimize} & g_5(\mathbf{b}) \\ \text{subject to} & g_1(\mathbf{b}) \leq 1 + \delta_P \\ & g_2(\mathbf{b}) \geq 1 - \delta_P \end{array} ,$$

with variable $\mathbf{b} \in \mathbb{R}^{M+1}$. This is identical to the problem

$$\begin{array}{ll} \text{minimize} & g_5(\mathbf{b}) \\ \text{subject to} & g_1(\mathbf{b}) - (1 + \delta_P) \leq 0 \\ & -g_2(\mathbf{b}) + (1 - \delta_P) \leq 0 \end{array} .$$

As the objective function $g_5(\mathbf{b})$ is quasiconvex and the constraints are convex, this problem is a quasiconvex optimization problem.

(c) Define the function $g_6(\mathbf{b})$ as follows.

$$g_6(\mathbf{b}) \triangleq \min \{k : b_{k+1} = \cdots = b_M = 0\} .$$

Note that $g_6(\mathbf{b})$ is a quasiconvex function of $\mathbf{b}$. To see this, note that the k_0-sublevel set $\mathcal{C}_{k_0}$ is given by

$$\mathcal{C}_{k_0} = \{\mathbf{b} : g_6(\mathbf{b}) \leq k_0\} = \{\mathbf{b} : b_{k_0+1} = \cdots = b_M = 0\} ,$$

which is an affine set. As such, $\mathcal{C}_{k_0}$ is convex and so $g_6(\mathbf{b})$ is quasiconvex.

With the function g_6 defined as such, the optimization problem becomes the following.

$$\begin{array}{ll} \text{minimize} & g_6(\mathbf{b}) \\ \text{subject to} & g_1(\mathbf{b}) \leq 1 + \delta_P \\ & g_2(\mathbf{b}) \geq 1 - \delta_P \\ & g_3(\mathbf{b}) \leq \delta_S \\ & g_4(\mathbf{b}) \geq -\delta_S \end{array} ,$$

with variable $\mathbf{b} \in \mathbb{R}^{M+1}$. This is identical to the following problem.

$$\begin{array}{ll} \text{minimize} & g_6(\mathbf{b}) \\ \text{subject to} & g_1(\mathbf{b}) - (1 + \delta_P) \leq 0 \\ & -g_2(\mathbf{b}) + (1 - \delta_P) \leq 0 \\ & g_3(\mathbf{b}) - \delta_S \leq 0 \\ & -g_4(\mathbf{b}) - \delta_S \leq 0 \end{array} ,$$

As the objective function $g_6(\mathbf{b})$ is quasiconvex and the constraints are convex, this problem is a quasiconvex optimization problem.

(d) After discretizing the frequency to $f_\ell = \ell / (2L)$, define the index sets $\mathcal{I}_P$ and $\mathcal{I}_S$ as follows.

$$\mathcal{I}_P \triangleq \{\ell : 0 \leq f_\ell \leq f_P\} , \ \mathcal{I}_S \triangleq \{\ell : f_S \leq f_\ell \leq 1/2\} .$$

In other words, $\mathcal{I}_P$ and $\mathcal{I}_S$ denote the sets of discretized frequency indices corresponding to the passband and stopband, respectively.

With these sets defined as such, the problem in part (c) can be solved via bisection by solving the following feasibility LP at each step.

$$\begin{array}{ll} \text{find} & \mathbf{b} \\ \text{subject to} & \mathbf{c}(f_\ell)^T \mathbf{b} - (1+\delta_P) \leq 0\,,\ \ell \in \mathcal{I}_P \\ & -\mathbf{c}(f_\ell)^T \mathbf{b} + (1-\delta_P) \leq 0\,,\ \ell \in \mathcal{I}_P \\ & \mathbf{c}(f_\ell)^T \mathbf{b} - \delta_S \leq 0\,,\ \ell \in \mathcal{I}_S \\ & -\mathbf{c}(f_\ell)^T \mathbf{b} - \delta_S \leq 0\,,\ \ell \in \mathcal{I}_S \end{array} .$$

Using `cvx` in MATLAB, we find $M^\star = 27$ and that the optimal filter which achieves this minimal filter length has a magnitude response in dB as shown in Figure 1. Close-ups of the passband and stopband regions are shown in Figure 2(a) and (b), respectively, and clearly satisfy the design specifications.

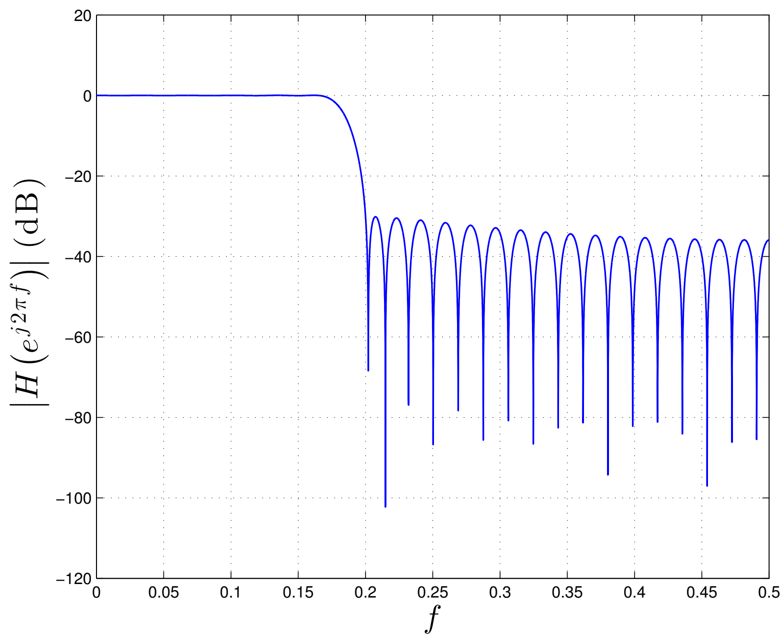
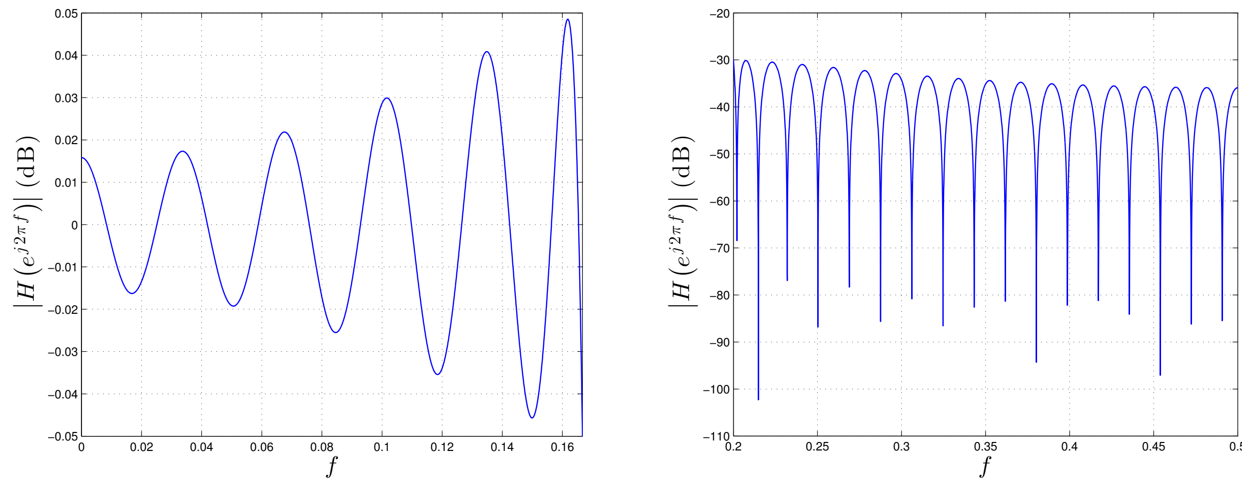


Figure 1: Minimal length FIR low-pass filter magnitude response in dB.

The following MATLAB code (employing `cvx`), was used to generate these results.

```
% Define filter specifications
f_P = 1./6;
f_S = 1./5;
delta_P_dB = 0.1;
delta_S_dB = -30;

% Map magnitude response specifications to amplitude response ones
K = 10.^(delta_P_dB./20);
delta_P = (K-1)./(K+1);
delta_S = 10.^(delta_S_dB./20);
```



(a) (b)

Figure 2: Minimal length FIR low-pass filter magnitude response in dB: (a) close-up of the passband region $0 \leq f \leq f_P$ and (b) close-up of the stopband region $f_S \leq f \leq 1/2$.

```
% Set discretized frequency parameters
L = 16383;
f = [0:L]./(2.*L);
I_P = find(f >= 0 & f <= f_P);
I_S = find(f >= f_S & f <= 0.5);

% Define bisection parameters
M_LL = 0;
M_UL = 100;
bisec_tol = 1.1;

% Carry out bisection using a while-loop
while (M_UL - M_LL >= bisec_tol)
    % Set the midpoint filter order
    M_MP = round((M_UL+M_LL)./2);
    % Compute the discretized cosine vectors
    k = (0:M_MP)';
    C = cos(2.*pi.*k*f);
    % Solve the feasibility LP
    cvx_begin
        variable b(M_MP + 1);
        subject to
            % Passband constraints
            C(:,I_P)'*b - (1+delta_P) <= 0;
            -C(:,I_P)'*b + (1-delta_P) <= 0;
            % Stopband constraints
            C(:,I_S)'*b - delta_S <= 0;
            -C(:,I_S)'*b - delta_S <= 0;
```

```
    cvx_end
    % Check for feasibility and update the bisection ignorance interval
    if (strcmp(cvx_status,'Solved'))
        M_UL = M_MP;
        b_opt = b;
        M_opt = M_MP;
    else
        M_LL = M_MP;
    end
end

% Plot the optimal magnitude response in dB
f = linspace(0,0.5,8192);
k = (0:M_opt)';
C = cos(2.*pi.*k*f);
H_R = C'*b_opt;
plot(f,20*log10(abs(H_R)),'LineWidth',1)
grid on
xlabel('$f$','Interpreter','LaTeX','FontSize',20)
ylabel('$\left| H\!\left( e^{j2\pi f} \right) \right|$ (dB)',...
    'Interpreter','LaTeX','FontSize',20)
```

2. (a) We get a subset $\mathcal{P} \subseteq \mathbb{R}^3$ (which will soon be shown to be a polyhedron) of locations $\mathbf{x}$ that are consistent with the camera measurements. To find the smallest box that covers *any* subset in $\mathbb{R}^3$, all we need to do is minimize and maximize the (linear) functions x_1, x_2, and x_3 to get $\mathbf{l}$ and $\mathbf{u}$. Here, $\mathcal{P}$ is a polyhedron, so we will end up solving 6 LPs, one to get each of l_1, l_2, l_3, u_1, u_2, and u_3.

To verify that $\mathcal{P}$ is a polyhedron, note that our measurements tell us that

$$\widehat{\mathbf{v}}_i - (\rho_i/2)\,\mathbf{1} \preceq \frac{1}{\mathbf{c}_i^T\mathbf{x} + d_i}\,(\mathbf{A}_i\mathbf{x} + \mathbf{b}_i) \preceq \widehat{\mathbf{v}}_i + (\rho_i/2)\,\mathbf{1}\,,\ i = 1,\ldots,m\,.$$

Multiplying through by $\left(\mathbf{c}_i^T\mathbf{x} + d_i\right)$, which is positive, we get

$$\left(\widehat{\mathbf{v}}_i - (\rho_i/2)\,\mathbf{1}\right)\left(\mathbf{c}_i^T\mathbf{x} + d_i\right) \preceq \mathbf{A}_i\mathbf{x} + \mathbf{b}_i \preceq \left(\widehat{\mathbf{v}}_i + (\rho_i/2)\,\mathbf{1}\right)\left(\mathbf{c}_i^T\mathbf{x} + d_i\right)\,,\ i = 1,\ldots,m\,,$$

which is a set of $2m$ linear inequalities in $\mathbf{x}$. In particular, it defines a set $\mathcal{P}$, which is a polyhedron.

To get l_k and u_k, we solve the LPs

$$\begin{array}{ll} \text{minimize/maximize} & x_k \\ \text{subject to} & \left(\widehat{\mathbf{v}}_i - (\rho_i/2)\,\mathbf{1}\right)\left(\mathbf{c}_i^T\mathbf{x} + d_i\right) \preceq \mathbf{A}_i\mathbf{x} + \mathbf{b}_i\,,\ i = 1,\ldots,m \\ & \mathbf{A}_i\mathbf{x} + \mathbf{b}_i \preceq \left(\widehat{\mathbf{v}}_i + (\rho_i/2)\,\mathbf{1}\right)\left(\mathbf{c}_i^T\mathbf{x} + d_i\right)\,,\ i = 1,\ldots,m \end{array},$$

for $k = 1, 2, 3$. Here, it is understood that l_k is found by solving the minimization problem, whereas u_k is found by solving the maximization problem.

(b) The following MATLAB script (using `cvx`) solves this specific problem instance.

```
% Load the data and extract the sections of the camera matrices
```

```
camera_data;
A1 = P1(1:2,1:3); b1 = P1(1:2,4); c1 = P1(3,1:3)'; d1 = P1(3,4);
A2 = P2(1:2,1:3); b2 = P2(1:2,4); c2 = P2(3,1:3)'; d2 = P2(3,4);
A3 = P3(1:2,1:3); b3 = P3(1:2,4); c3 = P3(3,1:3)'; d3 = P3(3,4);
A4 = P4(1:2,1:3); b4 = P4(1:2,4); c4 = P4(3,1:3)'; d4 = P4(3,4);

% Solve the 6 LPs to find the smallest bounding box consistent with the
% measurements
cvx_quiet(true);
for bounds = 1:6
    cvx_begin
    variable x(3);
    switch bounds
        case 1
            minimize x(1)
        case 2
            maximize x(1)
        case 3
            minimize x(2)
        case 4
            maximize x(2)
        case 5
            minimize x(3)
        case 6
            maximize x(3)
    end
    % Constraints for the 1st camera
    (vhat(:,1)-rho(1)/2)*(c1'*x + d1) <= A1*x + b1;
    A1*x + b1 <= (vhat(:,1)+rho(1)/2)*(c1'*x + d1);
    % Constraints for the 2nd camera
    (vhat(:,2)-rho(2)/2)*(c2'*x + d2) <= A2*x + b2;
    A2*x + b2 <= (vhat(:,2)+rho(2)/2)*(c2'*x + d2);
    % Constraints for the 3rd camera
    (vhat(:,3)-rho(3)/2)*(c3'*x + d3) <= A3*x + b3;
    A3*x + b3 <= (vhat(:,3)+rho(3)/2)*(c3'*x + d3);
    % Constraints for the 4th camera
    (vhat(:,4)-rho(4)/2)*(c4'*x + d4) <= A4*x + b4;
    A4*x + b4 <= (vhat(:,4)+rho(4)/2)*(c4'*x + d4);
    cvx_end
    val(bounds) = cvx_optval;
end

% Display the minimal bounding box bounds
disp(['l1 = ' num2str(val(1))]);
disp(['u1 = ' num2str(val(2))]);
disp(['l2 = ' num2str(val(3))]);
disp(['u2 = ' num2str(val(4))]);
disp(['l3 = ' num2str(val(5))]);
```

```
disp(['u3 = ' num2str(val(6))]);
```

The script returns the following results.

$$\begin{aligned} l_1 &= -0.99561\,, & u_1 &= -0.8245\,,\\ l_2 &= 0.27531\,, & u_2 &= 0.37837\,,\\ l_3 &= -0.67899\,, & u_3 &= -0.57352\,. \end{aligned}$$

3. (a) Consider the desired measurement equations

$$z_i = \phi^{-1}(y_i)\ ,\ i = 1, \ldots, m\,. \tag{1}$$

The function ϕ^{-1} is unknown and needs to be estimated here, however, we do have bounds on its derivative. Specifically, we have

$$1/\beta \le \left(\phi^{-1}\right)'(v) \le 1/\alpha\,, \tag{2}$$

for all v. To show this, note that we have

$$\phi\big(\phi^{-1}(v)\big) = v\,,\ \forall\ v\,.$$

Differentiating both sides of the above relation and using the chain rule, we have

$$\phi'(u)\left(\phi^{-1}\right)'(v) = 1\,,\ \text{where } u = \phi^{-1}(v)\ .$$

Hence, we get

$$\left(\phi^{-1}\right)'(v) = 1/\phi'(u)\ ,\ \text{where } u = \phi^{-1}(v)\ . \tag{3}$$

But, as $\alpha \le \phi'(u) \le \beta$ for all u, where $0 < \alpha < \beta$, we have

$$1/\beta \le 1/\phi'(u) \le 1/\alpha\,,\ \forall\ u\,.$$

Substituting this into (3) thus yields

$$1/\beta \le \left(\phi^{-1}\right)'(v) \le 1/\alpha\,,\ \forall\ v\,.$$

Returning to the problem at hand, note that from (1) and the mean value theorem, there is a $v \in (-\infty, \infty)$ for which we have

$$\left(\phi^{-1}\right)'(v) = \frac{z_{i+1} - z_i}{y_{i+1} - y_i}\,,$$

for any $i = 1, \ldots, m-1$. Substituting this into (2) and assuming that the data is given with y_i in nondecreasing order, we get

$$(1/\beta)\,(y_{i+1} - y_i) \le z_{i+1} - z_i \le (1/\alpha)\,(y_{i+1} - y_i)\ ,\ i = 1, \ldots, m-1\,.$$

Now, as $v_i = z_i - \mathbf{a}_i^T\mathbf{x}$ and v_i are i.i.d. with distribution $\mathcal{N}\big(0, \sigma^2\big)$ for $i = 1, \ldots, m$, the log-likelihood function $l(\mathbf{x}, \mathbf{z})$ has the form

$$l(\mathbf{x}, \mathbf{z}) = -C_1 \sum_{i=1}^{m} \left(z_i - \mathbf{a}_i^T\mathbf{x}\right)^2 + C_2\,,$$

where $C_1 \triangleq 1/\left(2\sigma^2\right)$ (which satisfies $C_1 > 0$) and $C_2 \triangleq (m/2)\log\left(2\pi\sigma^2\right)$ are constants. Thus, to find an ML estimate of $\mathbf{x}$ and $\mathbf{z}$, we can minimize the objective

$$f_0(\mathbf{x},\mathbf{z}) \triangleq \sum_{i=1}^{m} \left(z_i - \mathbf{a}_i^T\mathbf{x}\right)^2 = ||\mathbf{z} - \mathbf{A}\mathbf{x}||_2^2 \ ,$$

subject to the constraints. Note that here, the i-th row of $\mathbf{A}$ is $\mathbf{a}_i^T$ for $i = 1, \ldots, m$. This leads to the following problem.

$$\begin{array}{ll} \text{minimize} & ||\mathbf{z} - \mathbf{A}\mathbf{x}||_2^2 \\ \text{subject to} & (1/\beta)\left(y_{i+1} - y_i\right) \le z_{i+1} - z_i \le (1/\alpha)\left(y_{i+1} - y_i\right) \ , \ i = 1, \ldots, m-1 \end{array} \ ,$$

with variables $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^m$. However, this is a QP and hence a convex optimization problem. Note that the problem does not depend on the noise variance σ^2.

(b) The following MATLAB code, which invokes `cvx`, was used to solve this specific problem instance.

```
% Load the nonlinear measurement data
nonlin_meas_data;

% Generate the (m-1) x m first order difference matrix Delta_1
Delta_1_row = zeros(1,m);
Delta_1_row(1) = -1;
Delta_1_row(2) = 1;
Delta_1_col = zeros(1,m-1);
Delta_1_col(1) = -1;
Delta_1 = toeplitz(Delta_1_col,Delta_1_row);

% Solve the QP used to get the ML estimates of x and z using cvx
cvx_begin
    variable x(n);
    variable z(m);
    minimize norm(z - A*x)
    subject to
        (1./beta).*Delta_1*y <= Delta_1*z;
        Delta_1*z <= (1./alpha).*Delta_1*y;
cvx_end

% Display the ML estimate of x
disp('ML estimate of x:'); disp(x);

% Plot the estimated function \widehat{\phi}_{\mathrm{ml}}
plot(z,y,'LineWidth',1)
xlabel('$u$','Interpreter','LaTeX','FontSize',20)
ylabel('$\widehat{\phi}_{\mathrm{ml}}\!\left( u \right)$','Interpreter',...
    'LaTeX','FontSize',20)
```

This yields the ML estimate of $\mathbf{x}$ given as follows.

$$\widehat{\mathbf{x}}_{\text{ml}} = \left[\ 0.4819 \quad -0.4657 \quad 0.9364 \quad 0.9297 \ \right]^T \ .$$

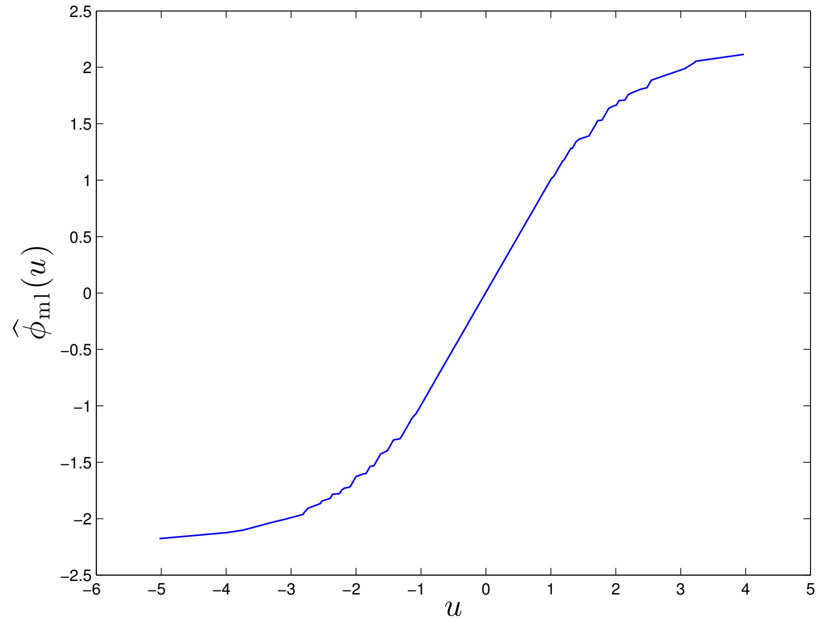


Figure 3: Plot of the ML estimate of the nonlinear function $\phi(u)$, namely $\widehat{\phi}_{\mathrm{ml}}(u)$. This was constructed by plotting $[\widehat{\mathbf{z}}_{\mathrm{ml}}]_i$ versus y_i for $i = 1, \ldots, m$.

A plot of the estimated function $\widehat{\phi}_{\mathrm{ml}}(u)$ is shown in Figure 3.

4. In this problem, to eliminate outliers successively, at each iteration, we compute the Löwner-John ellipsoid for the current set of data points and remove the point corresponding to the largest Lagrange multiplier and add it to the set of outliers. Let $\mathcal{D}_k$ and $\mathcal{O}_k$ denote, respectively, the data and outlier sets at the k-th iteration. Also, let $\mathbf{x}_i^{(k)}$ denote the point removed at the k-th iteration, i.e., the one that corresponds to the largest Lagrange multiplier. Then, we have the following ellipsoidal peeling algorithm.

Initialization:

$$\mathcal{D}_0 \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}\ ,\ \mathcal{O}_0 \triangleq \varnothing\,.$$

Iteration: For $k = 1, \ldots, N_{\mathrm{rem}}$, where N_{rem} denotes the total number of points to remove from the original data set, do the following.

1. Determine $\mathbf{x}_i^{(k)}$ as the point corresponding to the largest Lagrange multiplier of the following convex optimization problem.

$$\begin{array}{ll} \text{minimize} & -\log\det\mathbf{A} \\ \text{subject to} & ||\mathbf{A}\mathbf{x}_i + \mathbf{b}||_2 \le 1\,,\ \mathbf{x}_i \in \mathcal{D}_k \end{array} \iff \begin{array}{ll} \text{minimize} & -\left(\det\mathbf{A}\right)^{1/n} \\ \text{subject to} & ||\mathbf{A}\mathbf{x}_i + \mathbf{b}||_2 \le 1\,,\ \mathbf{x}_i \in \mathcal{D}_k \end{array}\,,$$

with variables $\mathbf{A} \in \mathbb{S}^n$ and $\mathbf{b} \in \mathbb{R}^n$. When it comes to solving instantiations of this problem using `cvx`, we will use the second equivalent form of the problem as the `det_rootn` subroutine is more efficient that the `log_det` one.

2. Update the data and outlier sets as follows.

$$\mathcal{D}_k = \mathcal{D}_{k-1} \setminus \left\{\mathbf{x}_i^{(k)}\right\}\ ,\ \mathcal{O}_k = \mathcal{O}_{k-1} \cup \left\{\mathbf{x}_i^{(k)}\right\}\,.$$

3. Increment k as $k \to k+1$.
4. If $k \leq N_{\mathrm{rem}}$, then go to 1. Otherwise, stop. The following MATLAB code, which uses `cvx`, was used to remove outliers from the given data set here.

```
% Load the data
ellip_peel_data;

% Set the number of outliers to remove
N_rem = 30;

% Initialize the ellipsoidal peeling algorithm
D = data;
O = [];
log_vol_array = [];
n = size(data,1);

% Run for-loop iteration to remove outliers successively
for k = 1:(N_rem+1)
    % Find the minimum volume ellipsoid containing all current data points
    cvx_begin
        variable A(n,n) symmetric;
        variable b(n);
        dual variable lambda;
        minimize (-det_rootn(A))
        %minimize (-log_det(A))
        subject to
            lambda : norms(A*D + repmat(b,1,size(D,2))) <= 1;
    cvx_end
    % Find the maximum dual variable (i.e., the largest Lagrange multiplier
    % and remove the corresponding data point
    [lambda_max,outlier_index] = max(lambda);
    % Update the outlier and data sets
    O = [O D(:,outlier_index)];
    D(:,outlier_index) = [];
    % Store the logarithm of the volume of the optimal ellipsoid
    log_vol_array = [log_vol_array (0.5.*log(1./det(A)))];
    clear A b;
end

% Plot the log of the volume of the optimal ellipsoid as a function of the
% number of points removed
figure
plot(0:N_rem,log_vol_array,'LineWidth',1)
grid on
xlabel('$\mathrm{card}\!\left( \mathcal{O} \right)$','Interpreter',...
    'LaTeX','FontSize',20)
ylabel('$\log\!\left(\mathrm{vol}\!\left(\mathcal{E}\right)\right)$',...
    'Interpreter','LaTeX','FontSize',20)
```

```
% Plot the data points to determine visually the number of outliers present
figure
plot(data(1,:),data(2,:),'.','LineWidth',1)
xlabel('$x$','Interpreter','LaTeX','FontSize',20)
ylabel('$y$','Interpreter','LaTeX','FontSize',20)
```

A plot of the number of points removed card($\mathcal{O}$) as a function of the logarithm of the volume of the resulting minimum volume covering ellipsoid $\mathcal{E}$ is shown in Figure 4(a). Along with this, a plot of the data points themselves is given in Figure 4(b).

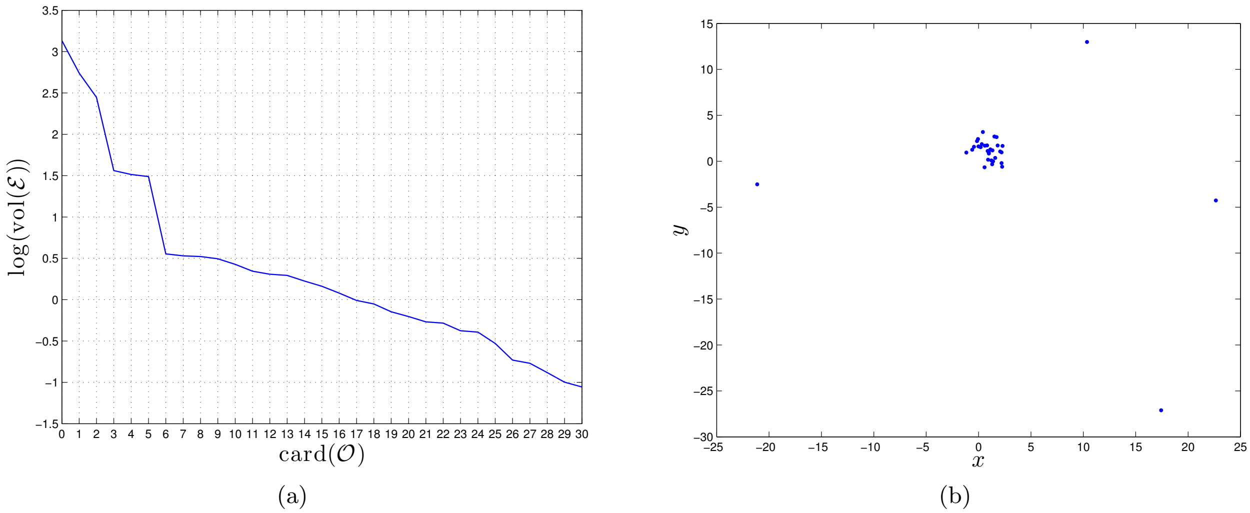


Figure 4: Ellipsoidal peeling plots: (a) logarithm of the volume of the minimum covering ellipsoid (i.e., log(vol($\mathcal{E}$))) versus the number of data points removed (i.e., card($\mathcal{O}$)) and (b) original data set.

From the 'eyeball' test, it is clear that there are 4 outliers. This is not immediately clear from the plot in Figure 4(a), which exhibits two knees. From Figure 4(a), it can be seen that there is a dramatic decrease in the volume of the minimum covering ellipsoid for the first 3 points and then there is some stalling for the next two points, which suggests that non-outlier points were erroneously added to $\mathcal{O}$. This corresponds to the first knee of the curve. Then, there is another dramatic decrease at the next point, followed by a slow and gradual shrinking of the volume from that point on. This corresponds to the second knee of the curve. If we add up only the number of points leading to a dramatic decrease of the ellipsoidal volume, we obtain 4 outliers, consistent with the 'eyeball' test.

The results of this simulation bring to light some of the advantages and disadvantages of this ellipsoidal peeling approach. If we accumulate only the points corresponding to large decreases in the ellipsoidal volume, then we obtain the correct number of outliers. However, the algorithm did not extract all the outliers before extracting some of the normal data points.

***5.** (a) Let $\boldsymbol{\nu} \in \mathbb{R}^n$ denote the dual variable. Then, the Lagrangian is given by the following.

$$\begin{aligned}
L(\mathbf{x}, \boldsymbol{\nu}) &= ||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 + \sum_{k=1}^{n} \nu_k \left(x_k^2 - 1 \right) , \\
&= \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} - 2\mathbf{b}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{b} + \mathbf{x}^T \left(\mathrm{diag}(\boldsymbol{\nu}) \right) \mathbf{x} - \mathbf{1}^T \boldsymbol{\nu} , \\
&= \mathbf{x}^T \left(\mathbf{A}^T \mathbf{A} + \mathrm{diag}(\boldsymbol{\nu}) \right) \mathbf{x} - 2\mathbf{b}^T \mathbf{A} \mathbf{x} - \mathbf{1}^T \boldsymbol{\nu} + \mathbf{b}^T \mathbf{b} .
\end{aligned} \tag{4}$$

From (4), it can be seen that $L(\mathbf{x}, \boldsymbol{\nu})$ is unbounded below in $\mathbf{x}$ if $\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu}) \not\succeq \mathbf{0}$ or $\left(-\mathbf{A}^T\mathbf{b}\right) \notin \mathcal{R}\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu})\right)$. When both $\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu}) \succeq \mathbf{0}$ and $\left(-\mathbf{A}^T\mathbf{b}\right) \in \mathcal{R}\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu})\right)$, the infimum of the Lagrangian occurs when the gradient with respect to $\mathbf{x}$ vanishes. This yields

$$\nabla L(\mathbf{x}, \boldsymbol{\nu}) = 2 \left(\mathbf{A}^T \mathbf{A} + \mathrm{diag}(\boldsymbol{\nu}) \right) \mathbf{x} - 2\mathbf{A}^T \mathbf{b} = \mathbf{0} ,$$

which is equivalent to the normal equations

$$\left(\mathbf{A}^T \mathbf{A} + \mathrm{diag}(\boldsymbol{\nu}) \right) \mathbf{x} = \mathbf{A}^T \mathbf{b} .$$

The solution to the normal equations is given by

$$\mathbf{x}^\star = \left(\mathbf{A}^T \mathbf{A} + \mathrm{diag}(\boldsymbol{\nu}) \right)^\# \mathbf{A}^T \mathbf{b} .$$

Substituting this into (4), it follows that the dual function $g(\boldsymbol{\nu}) = \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\nu})$ is given by

$$g(\boldsymbol{\nu}) = \begin{cases} -\mathbf{1}^T\boldsymbol{\nu} - \mathbf{b}^T\mathbf{A}\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu})\right)^\# \mathbf{A}^T\mathbf{b} + \mathbf{b}^T\mathbf{b} , & \mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu}) \succeq \mathbf{0} , \\ & \left(-\mathbf{A}^T\mathbf{b}\right) \in \mathcal{R}\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu})\right) \\ -\infty , & \text{otherwise} \end{cases} .$$

Hence, the dual problem is the following.

$$\begin{array}{ll} \text{maximize} & -\mathbf{1}^T\boldsymbol{\nu} - \mathbf{b}^T\mathbf{A}\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu})\right)^\# \mathbf{A}^T\mathbf{b} + \mathbf{b}^T\mathbf{b} \\ \text{subject to} & \mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu}) \succeq \mathbf{0} \\ & \left(-\mathbf{A}^T\mathbf{b}\right) \in \mathcal{R}\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu})\right) \end{array} .$$

Expressing the problem in epigraph form yields the following equivalent problem.

$$\begin{array}{ll} \text{maximize} & -\mathbf{1}^T\boldsymbol{\nu} - t + \mathbf{b}^T\mathbf{b} \\ \text{subject to} & \mathbf{b}^T\mathbf{A}\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu})\right)^\# \mathbf{A}^T\mathbf{b} \le t \\ & \mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu}) \succeq \mathbf{0} \\ & \left(-\mathbf{A}^T\mathbf{b}\right) \in \mathcal{R}\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu})\right) \end{array} .$$

In turn, this is identical to the problem

$$\begin{array}{ll} \text{maximize} & -\mathbf{1}^T\boldsymbol{\nu} - t + \mathbf{b}^T\mathbf{b} \\ \text{subject to} & t - \left(-\mathbf{b}^T\mathbf{A}\right)\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu})\right)^\# \left(-\mathbf{A}^T\mathbf{b}\right) \ge 0 \\ & \mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu}) \succeq \mathbf{0} \\ & \left(-\mathbf{A}^T\mathbf{b}\right) \in \mathcal{R}\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}(\boldsymbol{\nu})\right) \end{array} ,$$

which, upon using Schur complements, leads to the following equivalent problem, which is an SDP.

$$\begin{array}{ll} \text{maximize} & -\mathbf{1}^T\boldsymbol{\nu} - t + \mathbf{b}^T\mathbf{b} \\ \text{subject to} & \begin{bmatrix} \mathbf{A}^T\mathbf{A} + \operatorname{diag}(\boldsymbol{\nu}) & -\mathbf{A}^T\mathbf{b} \\ -\mathbf{b}^T\mathbf{A} & t \end{bmatrix} \succeq \mathbf{0} \end{array}.$$

(b) We first write the dual problem as a minimization problem by negating the objective function. This leads to the problem

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T\boldsymbol{\nu} + t - \mathbf{b}^T\mathbf{b} \\ \text{subject to} & \begin{bmatrix} \mathbf{A}^T\mathbf{A} + \operatorname{diag}(\boldsymbol{\nu}) & -\mathbf{A}^T\mathbf{b} \\ -\mathbf{b}^T\mathbf{A} & t \end{bmatrix} \succeq \mathbf{0} \end{array}.$$

To handle the linear matrix inequality (LMI) constraint, we introduce a Lagrange multiplier matrix of the form

$$\begin{bmatrix} \mathbf{Z} & \mathbf{z} \\ \mathbf{z}^T & \lambda \end{bmatrix}.$$

With this, the Lagrangian is given by the following.

$$\begin{aligned} L(\boldsymbol{\nu}, t, \mathbf{Z}, \mathbf{z}, \lambda) &= \mathbf{1}^T\boldsymbol{\nu} + t - \mathbf{b}^T\mathbf{b} - \operatorname{tr}\left(\begin{bmatrix} \mathbf{Z} & \mathbf{z} \\ \mathbf{z}^T & \lambda \end{bmatrix}\begin{bmatrix} \mathbf{A}^T\mathbf{A} + \operatorname{diag}(\boldsymbol{\nu}) & -\mathbf{A}^T\mathbf{b} \\ -\mathbf{b}^T\mathbf{A} & t \end{bmatrix}\right), \\ &= \mathbf{1}^T\boldsymbol{\nu} + t - \mathbf{b}^T\mathbf{b} - \operatorname{tr}\big(\mathbf{Z}\left(\mathbf{A}^T\mathbf{A} + \operatorname{diag}(\boldsymbol{\nu})\right) - \mathbf{z}\mathbf{b}^T\mathbf{A}\big) - \big(-\mathbf{z}^T\mathbf{A}^T\mathbf{b} + \lambda t\big), \\ &= \mathbf{1}^T\boldsymbol{\nu} + t - \mathbf{b}^T\mathbf{b} - \operatorname{tr}\big(\mathbf{A}^T\mathbf{A}\mathbf{Z}\big) - \operatorname{diag}(\mathbf{Z})^T\boldsymbol{\nu} + \mathbf{b}^T\mathbf{A}\mathbf{z} + \mathbf{z}^T\mathbf{A}^T\mathbf{b} - \lambda t, \\ &= (\mathbf{1} - \operatorname{diag}(\mathbf{Z}))^T\boldsymbol{\nu} + t(1-\lambda) - \operatorname{tr}\big(\mathbf{A}^T\mathbf{A}\mathbf{Z}\big) + 2\mathbf{b}^T\mathbf{A}\mathbf{z} - \mathbf{b}^T\mathbf{b}. \end{aligned}$$

This is unbounded below unless $\operatorname{diag}(\mathbf{Z}) = \mathbf{1}$ and $\lambda = 1$. As such, the dual function is given by

$$g(\mathbf{Z}, \mathbf{z}, \lambda) = \begin{cases} -\operatorname{tr}\big(\mathbf{A}^T\mathbf{A}\mathbf{Z}\big) + 2\mathbf{b}^T\mathbf{A}\mathbf{z} - \mathbf{b}^T\mathbf{b}, & \operatorname{diag}(\mathbf{Z}) = \mathbf{1},\ \lambda = 1 \\ -\infty & \text{otherwise} \end{cases}.$$

Thus, the dual problem of the SDP of part (a) is the following.

$$\begin{array}{ll} \text{maximize} & -\operatorname{tr}\big(\mathbf{A}^T\mathbf{A}\mathbf{Z}\big) + 2\mathbf{b}^T\mathbf{A}\mathbf{z} - \mathbf{b}^T\mathbf{b} \\ \text{subject to} & \operatorname{diag}(\mathbf{Z}) = \mathbf{1} \\ & \begin{bmatrix} \mathbf{Z} & \mathbf{z} \\ \mathbf{z}^T & 1 \end{bmatrix} \succeq \mathbf{0} \end{array}.$$

Expressing this maximization problem as a minimization problem by negating the objective leads to the following equivalent problem.

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}\big(\mathbf{A}^T\mathbf{A}\mathbf{Z}\big) - 2\mathbf{b}^T\mathbf{A}\mathbf{z} + \mathbf{b}^T\mathbf{b} \\ \text{subject to} & \operatorname{diag}(\mathbf{Z}) = \mathbf{1} \\ & \begin{bmatrix} \mathbf{Z} & \mathbf{z} \\ \mathbf{z}^T & 1 \end{bmatrix} \succeq \mathbf{0} \end{array}. \tag{5}$$

This is the desired form here.

To see that (5) is a relaxation of the original problem, note that the binary least-squares problem is equivalent to

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}\big(\mathbf{A}^T\mathbf{A}\mathbf{Z}\big) - 2\mathbf{b}^T\mathbf{A}\mathbf{z} + \mathbf{b}^T\mathbf{b} \\ \text{subject to} & \operatorname{diag}(\mathbf{Z}) = \mathbf{1} \\ & \mathbf{Z} = \mathbf{z}\mathbf{z}^T \end{array}. \tag{6}$$

Suppose we relax the equality constraint $\mathbf{Z} = \mathbf{z}\mathbf{z}^T$ with the weaker inequality constraint $\mathbf{Z} \succeq \mathbf{z}\mathbf{z}^T$. Using Schur complements, this weaker inequality constraint is equivalent to

$$\begin{bmatrix} 1 & \mathbf{z}^T \\ \mathbf{z} & \mathbf{Z} \end{bmatrix} \succeq \mathbf{0}.$$

But note that we have the following.

$$\begin{aligned} \begin{bmatrix} 1 & \mathbf{z}^T \\ \mathbf{z} & \mathbf{Z} \end{bmatrix} \succeq \mathbf{0} \quad &\Longleftrightarrow \quad \begin{bmatrix} \mathbf{0}_{n\times 1} & \mathbf{I}_n \\ 1 & \mathbf{0}_{1\times n} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{z}^T \\ \mathbf{z} & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{1\times n} & 1 \\ \mathbf{I}_n & \mathbf{0}_{n\times 1} \end{bmatrix} \succeq \mathbf{0}, \\ &\Longleftrightarrow \quad \begin{bmatrix} \mathbf{Z} & \mathbf{z} \\ \mathbf{z}^T & 1 \end{bmatrix} \succeq \mathbf{0}. \end{aligned}$$

Substituting this equivalent weaker inequality constraint into (6), we get the problem from (5).

If we have

$$\operatorname{rank}\left(\begin{bmatrix} \mathbf{Z} & \mathbf{z} \\ \mathbf{z}^T & 1 \end{bmatrix}\right) = 1$$

at the optimum of (5), which is equivalent to saying that $\mathbf{Z} = \mathbf{z}\mathbf{z}^T$, then we satisfy the equality constraint $\mathbf{Z} = \mathbf{z}\mathbf{z}^T$ of (6), and so the relaxation is exact in this case. Thus, the optimal values of problems (6) and (5) are equal, and the optimal solution $\mathbf{z}$ of (5) is optimal for (6).

(c) Consider the problem

$$\begin{array}{ll} \text{minimize} & E\Big[||\mathbf{A}\mathbf{v} - \mathbf{b}||_2^2\Big] \\ \text{subject to} & E\big[v_k^2\big] = 1\,,\ k = 1, \ldots, n \end{array}. \tag{7}$$

Note that we have the following upon expanding the objective.

$$\begin{aligned} E\Big[||\mathbf{A}\mathbf{v} - \mathbf{b}||_2^2\Big] &= E\Big[(\mathbf{A}\mathbf{v} - \mathbf{b})^T (\mathbf{A}\mathbf{v} - \mathbf{b})\Big]\,, \\ &= E\Big[\operatorname{tr}\Big((\mathbf{A}\mathbf{v} - \mathbf{b})(\mathbf{A}\mathbf{v} - \mathbf{b})^T\Big)\Big]\,, \\ &= \operatorname{tr}\Big(E\Big[(\mathbf{A}\mathbf{v} - \mathbf{b})(\mathbf{A}\mathbf{v} - \mathbf{b})^T\Big]\Big)\,, \\ &= \operatorname{tr}\big(E\big[\mathbf{A}\mathbf{v}\mathbf{v}^T\mathbf{A}^T - \mathbf{A}\mathbf{v}\mathbf{b}^T - \mathbf{b}\mathbf{v}^T\mathbf{A}^T + \mathbf{b}\mathbf{b}^T\big]\big)\,, \\ &= \operatorname{tr}\Big(\mathbf{A}\left(E\big[\mathbf{v}\mathbf{v}^T\big]\right)\mathbf{A}^T - \mathbf{A}\left(E[\mathbf{v}]\right)\mathbf{b}^T - \mathbf{b}\left(E[\mathbf{v}]\right)^T\mathbf{A}^T + \mathbf{b}\mathbf{b}^T\Big)\,, \\ &= \operatorname{tr}\big(\mathbf{A}\mathbf{Z}\mathbf{A}^T - \mathbf{A}\mathbf{z}\mathbf{b}^T - \mathbf{b}\mathbf{z}^T\mathbf{A}^T + \mathbf{b}\mathbf{b}^T\big)\,, &(8) \\ &= \operatorname{tr}\big(\mathbf{A}^T\mathbf{A}\mathbf{Z}\big) - \mathbf{b}^T\mathbf{A}\mathbf{z} - \mathbf{z}^T\mathbf{A}^T\mathbf{b} + \mathbf{b}^T\mathbf{b}\,, \\ &= \operatorname{tr}\big(\mathbf{A}^T\mathbf{A}\mathbf{Z}\big) - 2\mathbf{b}^T\mathbf{A}\mathbf{z} + \mathbf{b}^T\mathbf{b}\,. &(9) \end{aligned}$$

Here, (8) follows from the fact that $\mathbf{z} = E[\mathbf{v}]$ and $\mathbf{Z} = E\big[\mathbf{v}\mathbf{v}^T\big]$. Similarly, for the left-hand-side of the equality constraints of (7), we get the following.

$$E\big[v_k^2\big] = Z_{k,k} = [\mathbf{Z}]_{k,k} \ , \ k = 1, \ldots, n \, . \tag{10}$$

Finally, the covariance matrix of $\mathbf{v}$, given by

$$\mathbf{C}_{\mathbf{v}} \triangleq E\Big[(\mathbf{v} - E[\mathbf{v}])\,(\mathbf{v} - E[\mathbf{v}])^T\Big] = \mathbf{Z} - \mathbf{z}\mathbf{z}^T \, ,$$

must be positive semidefinite, i.e., $\mathbf{C}_{\mathbf{v}} \succeq \mathbf{0}$. Hence, we must have $\mathbf{Z} - \mathbf{z}\mathbf{z}^T \succeq \mathbf{0}$, which was found in part (b) to be equivalent to

$$\begin{bmatrix} \mathbf{Z} & \mathbf{z} \\ \mathbf{z}^T & 1 \end{bmatrix} \succeq \mathbf{0} \, .$$

Combining this constraint with the results of (9) and (10), it follows that the problem from (7) is equivalent to the problem

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}\big(\mathbf{A}^T\mathbf{A}\mathbf{Z}\big) - 2\mathbf{b}^T\mathbf{A}\mathbf{z} + \mathbf{b}^T\mathbf{b} \\ \text{subject to} & \operatorname{diag}(\mathbf{Z}) = \mathbf{1} \\ & \begin{bmatrix} \mathbf{Z} & \mathbf{z} \\ \mathbf{z}^T & 1 \end{bmatrix} \succeq \mathbf{0} \end{array} ,$$

which is the same as (5) from part (b).

(d) The following MATLAB code, employing `cvx`, was used to compute suboptimal feasible solutions using the desired heuristics.

```
% Initialize the problem data
randn('state',0)
s = 0.5;
%s = 1;
%s = 2;
%s = 3;
m = 50;
n = 40;
A = randn(m,n);
xhat = sign(randn(n,1));
b = A*xhat + s.*randn(m,1);
f_xhat = norm(A*xhat - b).^2;

% (i) Compute the sign of the solution to the LS problem
x_a = sign(A\b);
f_x_a = norm(A*x_a - b).^2;

% (ii) Compute the sign of the solution to the SDP of part (b)
%cvx_precision high
cvx_begin sdp
    variable z(n);
```

```
    variable Z(n,n) symmetric;
    minimize (trace(A'*A*Z) - 2*b'*A*z + b'*b)
    subject to
        [Z z ; z' 1] >= 0;
        diag(Z) == 1;
cvx_end
x_b = sign(z);
f_x_b = norm(A*x_b - b).^2;

% (iii) Compute the sign of the rank-one approximation of the optimal
% solution of the SDP of part (b)
Y = [Z z ; z' 1];
[V,D] = eig(Y);
[eig_sorted,eig_index_sorted] = sort(diag(D),'descend');
v_1 = V(1:n,eig_index_sorted(1));
x_c = sign(z);
f_x_c = norm(A*x_c - b).^2;

% (iv) Compute the sign of the random sample with mean z and second moment
% Z
N = 100;
U = randn(n,N);
V_tilde = sqrtm(Z - z*z')*U + repmat(z,1,N);
X_d = sign(real(V_tilde));
R = A*X_d - repmat(b,1,N);
F_x_d = sum(R.^2,1);
[f_x_d,x_d_index] = min(F_x_d);
x_d = X_d(:,x_d_index);

% Report the objective function values for each approach
disp(['f_xhat = ' num2str(f_xhat)]);
disp(['f_x_a = ' num2str(f_x_a)]);
disp(['f_x_b = ' num2str(f_x_b)]);
disp(['f_x_c = ' num2str(f_x_c)]);
disp(['f_x_d = ' num2str(f_x_d)]);
disp(['SDP lower bound = ' num2str(cvx_optval)]);

% Report the square of the l_2-norm of the difference between the actual
% input to the problem and each heuristic solution
% solutions and the actual input to the problem
disp(['||xhat - x_a||_{2}^2 = ' num2str(norm(xhat-x_a).^2)]);
disp(['||xhat - x_b||_{2}^2 = ' num2str(norm(xhat-x_b).^2)]);
disp(['||xhat - x_c||_{2}^2 = ' num2str(norm(xhat-x_c).^2)]);
disp(['||xhat - x_d||_{2}^2 = ' num2str(norm(xhat-x_d).^2)]);
```

To generate samples of $\mathcal{N}\big(\mathbf{z}, \mathbf{Z} - \mathbf{z}\mathbf{z}^T\big)$, we used the following result from probability theory. If $\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $\mathbf{w} \triangleq \mathbf{B}\mathbf{u} + \mathbf{c}$ is such that $\mathbf{w} \sim \mathcal{N}\big(\mathbf{B}\boldsymbol{\mu} + \mathbf{c}, \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^T\big)$. Thus, if $\mathbf{u} \sim \mathcal{N}(\mathbf{0}_{n\times 1}, \mathbf{I}_n)$, then $\mathbf{w} \triangleq (\mathbf{Z} - \mathbf{z}\mathbf{z})^{1/2}\,\mathbf{u} + \mathbf{z}$ is such that $\mathbf{w} \sim \mathcal{N}\big(\mathbf{z}, \mathbf{Z} - \mathbf{z}\mathbf{z}^T\big)$.

The following table lists, for each s, the values of

$$f(\mathbf{x}) = ||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 \ ,$$

for $\mathbf{x} = \widehat{\mathbf{x}}$ and the four suboptimal solutions (i)-(iv), along with the lower bound obtained from the SDP relaxation given in (5).

s	$f(\widehat{\mathbf{x}})$	$f\big(\mathbf{x}^{(\mathrm{a})}\big)$	$f\big(\mathbf{x}^{(\mathrm{b})}\big)$	$f\big(\mathbf{x}^{(\mathrm{c})}\big)$	$f\big(\mathbf{x}^{(\mathrm{d})}\big)$	SDP lower bound
0.5	17.3243	17.3243	17.3243	17.3243	17.3243	16.4217
1	69.2974	162.0505	69.2974	69.2974	69.2974	61.9026
2	277.1895	908.5323	277.1895	277.1895	277.1895	230.4446
3	623.6765	1151.5117	673.6883	673.6883	623.6765	489.0238

From this table and the results of the simulations carried out here, we make the following observations.

- For $s = 0.5$, all heuristics return $\widehat{\mathbf{x}}$. This is likely to be the global optimum, but that is not necessarily true. However, from the lower bound, we know that the global optimum is in the interval $[16.4217, 17.3243]$, so even if 17.3243 is not the global optimum, it is quite close.
- For higher values of s, the result from the first heuristic $\mathbf{x}^{(\mathrm{a})}$ is substantially worse than those derived from the SDP-based heuristics.
- All three SDP-based heuristics return $\widehat{\mathbf{x}}$ for $s = 1$ and $s = 2$.
- For $s = 3$, the randomized rounding method returns $\widehat{\mathbf{x}}$. The other SDP-based heuristics give slightly higher values.