CVX101 Homework 5 solutions

6.2 *ℓ_1-, ℓ_2-, and ℓ_∞-norm approximation by a constant vector.* What is the solution of the norm approximation problem with one scalar variable $x \in \mathbf{R}$,

$$\text{minimize} \quad \|x\mathbf{1} - b\|,$$

for the ℓ_1-, ℓ_2-, and ℓ_∞-norms?

Solution.

(a) ℓ_2-norm: the average $\mathbf{1}^T b/m$.

(b) ℓ_1-norm: the (or a) median of the coefficients of b.

(c) ℓ_∞-norm: the midrange point $(\min b_i + \max b_i)/2$.

A5.13 *Fitting with censored data.* In some experiments there are two kinds of measurements or data available: The usual ones, in which you get a number (say), and *censored data*, in which you don't get the specific number, but are told something about it, such as a lower bound. A classic example is a study of lifetimes of a set of subjects (say, laboratory mice). For those who have died by the end of data collection, we get the lifetime. For those who have not died by the end of data collection, we do not have the lifetime, but we do have a lower bound, *i.e.*, the length of the study. These are the censored data values.

We wish to fit a set of data points,

$$(x^{(1)}, y^{(1)}), \ldots, (x^{(K)}, y^{(K)}),$$

with $x^{(k)} \in \mathbf{R}^n$ and $y^{(k)} \in \mathbf{R}$, with a linear model of the form $y \approx c^T x$. The vector $c \in \mathbf{R}^n$ is the model parameter, which we want to choose. We will use a least-squares criterion, *i.e.*, choose c to minimize

$$J = \sum_{k=1}^{K} \left(y^{(k)} - c^T x^{(k)} \right)^2.$$

Here is the tricky part: some of the values of $y^{(k)}$ are censored; for these entries, we have only a (given) lower bound. We will re-order the data so that $y^{(1)}, \ldots, y^{(M)}$ are given (*i.e.*, uncensored), while $y^{(M+1)}, \ldots, y^{(K)}$ are all censored, *i.e.*, unknown, but larger than D, a given number. All the values of $x^{(k)}$ are known.

(a) Explain how to find c (the model parameter) and $y^{(M+1)}, \ldots, y^{(K)}$ (the censored data values) that minimize J.

(b) Carry out the method of part (a) on the data values in `cens_fit_data.m`. Report $\hat{c}$, the value of c found using this method.

Also find $\hat{c}_{\rm ls}$, the least-squares estimate of c obtained by simply ignoring the censored data samples, *i.e.*, the least-squares estimate based on the data

$$(x^{(1)}, y^{(1)}), \ldots, (x^{(M)}, y^{(M)}).$$

The data file contains $c_{\rm true}$, the true value of c, in the vector `c_true`. Use this to give the two relative errors

$$\frac{\|c_{\rm true} - \hat{c}\|_2}{\|c_{\rm true}\|_2}, \qquad \frac{\|c_{\rm true} - \hat{c}_{\rm ls}\|_2}{\|c_{\rm true}\|_2}.$$

Solution.

(a) The trick is to introduce dummy variables to serve as placeholders for the measurements which are censored, *i.e.*, $y^{(k)}$, $k = M+1, \ldots, K$. By introducing the dummy variables $(z^{(1)}, \ldots, z^{(K-M)})$, we get the QP

$$\begin{array}{ll} \mbox{minimize} & \sum_{k=1}^{M} \left(y^{(k)} - c^T x^{(k)}\right)^2 + \sum_{k=M+1}^{K} \left(z^{(k-M)} - c^T x^{(k)}\right)^2 \\ \mbox{subject to} & z^{(k)} \geq D, \quad k = 1, \ldots, K-M, \end{array}$$

where the variables are c and $z^{(k)}$, $k = 1, \ldots, K-M$.

(b) The following code solves the problem

```
cens_fit_data;

% Using censored data method
cvx_begin
    variables c(n) z(K-M)
    minimize(sum_square(y-X(:,1:M)'*c)+sum_square(z-X(:,M+1:K)'*c))
    subject to
        z >= D
cvx_end
c_cens = c;

% Comparison to least squares method, ignoring all censored data
cvx_begin
    variable c(n)
    minimize(sum_square(y-X(:,1:M)'*c))
cvx_end
c_ls = c;

[c_true c_cens c_ls]
cens_relerr = norm(c_cens-c_true)/norm(c_true)
ls_relerr = norm(c_ls-c_true)/norm(c_true)
```

We get the following estimates of our parameter vector:

```
[c_true c_cens c_ls] =
   -0.4326   -0.2946   -0.3476
   -1.6656   -1.7541   -1.7955
    0.1253    0.2589    0.2000
    0.2877    0.2241    0.1672
   -1.1465   -0.9917   -0.8357
    1.1909    1.3018    1.3005
    1.1892    1.4262    1.8276
   -0.0376   -0.1554   -0.5612
    0.3273    0.3785    0.3686
    0.1746    0.2261   -0.0454
   -0.1867   -0.0826   -0.1096
    0.7258    1.0427    1.5265
   -0.5883   -0.4648   -0.4980
    2.1832    2.1942    2.4164
   -0.1364   -0.3586   -0.5563
    0.1139   -0.1973   -0.3701
    1.0668    1.0194    0.9900
    0.0593   -0.1186   -0.2539
   -0.0956   -0.1211   -0.1762
   -0.8323   -0.7523   -0.4349
```

This gives a relative error of 0.1784 for $\hat{c}$, and a relative error of 0.3907 for $\hat{c}_{\text{ls}}$.

A5.2 *Minimax rational fit to the exponential.* (See exercise 6.9 of *Convex Optimization.*) We consider the specific problem instance with data

$$t_i = -3 + 6(i-1)/(k-1), \quad y_i = e^{t_i}, \quad i = 1, \ldots, k,$$

where $k = 201$. (In other words, the data are obtained by uniformly sampling the exponential function over the interval $[-3, 3]$.) Find a function of the form

$$f(t) = \frac{a_0 + a_1 t + a_2 t^2}{1 + b_1 t + b_2 t^2}$$

that minimizes $\max_{i=1,\ldots,k} |f(t_i) - y_i|$. (We require that $1 + b_1 t_i + b_2 t_i^2 > 0$ for $i = 1, \ldots, k$.)

Find optimal values of $a_0,\ a_1,\ a_2,\ b_1,\ b_2$, and give the optimal objective value, computed to an accuracy of 0.001. Plot the data and the optimal rational function fit on the same plot. On a different plot, give the fitting error, *i.e.*, $f(t_i) - y_i$.

Hint. You can use `strcmp(cvx_status,'Solved')`, after `cvx_end`, to check if a feasibility problem is feasible.

Solution. The objective function (and therefore also the problem) is not convex, but it is quasiconvex. We have $\max_{i=1,\ldots,k} |f(t_i) - y_i| \leq \gamma$ if and only if

$$\left| \frac{a_0 + a_1 t_i + a_2 t_i^2}{1 + b_1 t_i + b_2 t_i^2} - y_i \right| \leq \gamma, \quad i = 1, \ldots, k.$$

This is equivalent to (since the denominator is positive)

$$|a_0 + a_1 t + a_2 t_i^2 - y_i(1 + b_1 t_i + b_2 t_i^2)| \leq \gamma(1 + b_1 t_i + b_2 t_i^2), \quad i = 1, \ldots, k,$$

which is a set of $2k$ linear inequalities in the variables a and b (for fixed γ). In particular, this shows the objective is quasiconvex. (In fact, it is a generalized linear fractional function.)

To solve the problem we can use a bisection method, solving an LP feasibility problem at each step. At each step we select some value of γ and solve the feasibility problem

$$\begin{array}{ll} \text{find} & a,\ b \\ \text{subject to} & |a_0 + a_1 t_i + a_2 t_i^2 - y_i(1 + b_1 t_i + b_2 t_i^2)| \leq \gamma(1 + b_1 t_i + b_2 t_i^2), \quad i = 1, \ldots, k, \end{array}$$

with variables a and b. (Note that as long as $\gamma > 0$, the condition that the denominator is positive is enforced automatically.) This can be turned into the LP feasibility problem

$$\begin{array}{ll} \text{find} & a,\ b \\ \text{subject to} & a_0 + a_1 t_i + a_2 t_i^2 - y_i(1 + b_1 t_i + b_2 t_i^2) \leq \gamma(1 + b_1 t_i + b_2 t_i^2), \quad i = 1, \ldots, k \\ & a_0 + a_1 t_i + a_2 t_i^2 - y_i(1 + b_1 t_i + b_2 t_i^2) \geq -\gamma(1 + b_1 t_i + b_2 t_i^2), \quad i = 1, \ldots, k. \end{array}$$

The following Matlab code solves the problem for the particular problem instance.

```
k=201;
t=(-3:6/(k-1):3)';
y=exp(t);

Tpowers=[ones(k,1) t t.^2];

u=exp(3); l=0; % initial upper and lower bounds
bisection_tol=1e-3; % bisection tolerance

while u-l>= bisection_tol
    gamma=(l+u)/2;
    cvx_begin % solve the feasibility problem
    cvx_quiet(true);
    variable a(3);
    variable b(2);
```

```
    subject to
        abs(Tpowers*a-y.*(Tpowers*[1;b])) <= gamma*Tpowers*[1;b];
    cvx_end

    if strcmp(cvx_status,'Solved')
        u=gamma;
        a_opt=a;
        b_opt=b;
        objval_opt=gamma;
    else
        l=gamma;
    end
end

y_fit=Tpowers*a_opt./(Tpowers*[1;b_opt]);

figure(1);
plot(t,y,'b', t,y_fit,'r+');
xlabel('t');
ylabel('y');

figure(2);
plot(t, y_fit-y);
xlabel('t');
ylabel('err');
```

The optimal values are

$$a_0 = 1.0099, \quad a_1 = 0.6117, \quad a_2 = 0.1134, \quad b_1 = -0.4147, \quad b_2 = 0.0485,$$

and the optimal objective value is 0.0233. We also get the following plots.

A6.6 *Maximum likelihood estimation of an increasing nonnegative signal.* We wish to estimate a scalar signal $x(t)$, for $t = 1, 2, \ldots, N$, which is known to be nonnegative and monotonically nondecreasing:

$$0 \leq x(1) \leq x(2) \leq \cdots \leq x(N).$$

This occurs in many practical problems. For example, $x(t)$ might be a measure of wear or deterioration, that can only get worse, or stay the same, as time t increases. We are also given that $x(t) = 0$ for $t \leq 0$.

We are given a noise-corrupted moving average of x, given by

$$y(t) = \sum_{\tau=1}^{k} h(\tau)x(t-\tau) + v(t), \quad t = 2, \ldots, N+1,$$

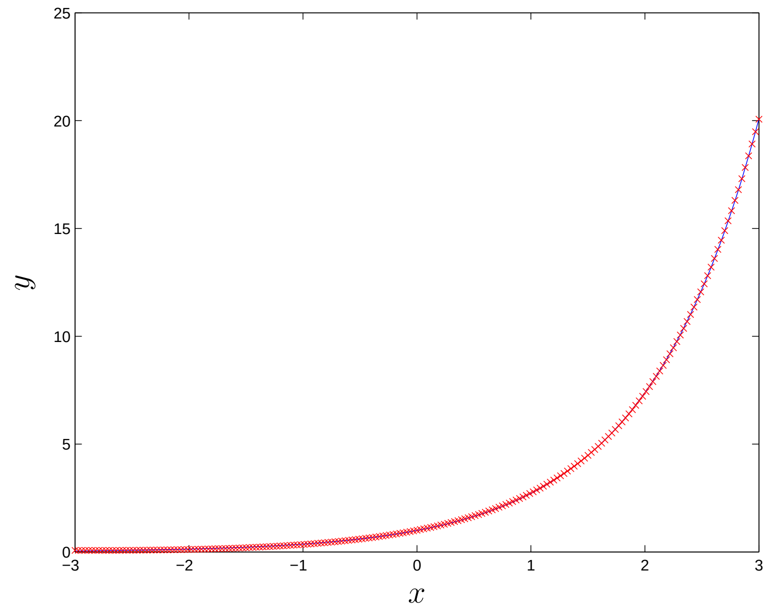


Figure 1 Chebyshev fit with rational function. The line represents the data and the crosses the fitted points.

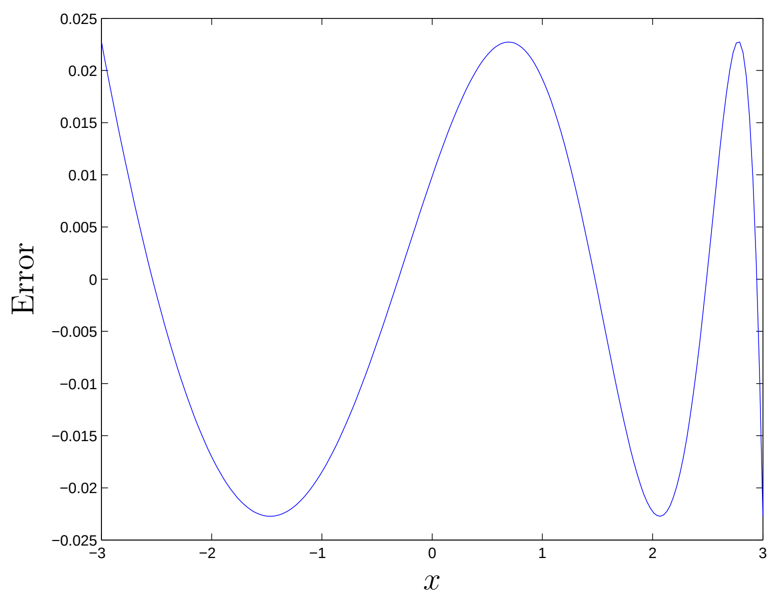


Figure 2 Fitting error for Chebyshev fit of exponential with rational function.

where $v(t)$ are independent $\mathcal{N}(0,1)$ random variables.

(a) Show how to formulate the problem of finding the maximum likelihood estimate of x, given y, taking into account the prior assumption that x is nonnegative and monotonically nondecreasing, as a convex optimization problem. Be sure to indicate what the problem variables are, and what the problem data are.

(b) We now consider a specific instance of the problem, with problem data (*i.e.*, N, k, h, and y) given in the file `ml_estim_incr_signal_data.m`. (This file contains the true signal `xtrue`, which of course you cannot use in creating your estimate.) Find the maximum likelihood estimate $\hat{x}_{\rm ml}$, and plot it, along with the true signal. Also find and plot the maximum likelihood estimate $\hat{x}_{\rm ml,free}$ *not taking into account the signal nonnegativity and monotonicity.*

   Hint. The function `conv` (convolution) is overloaded to work with CVX.

Solution.

(a) To simplify our notation, we let the signal $\hat{y}_x$ be the noiseless moving average of the signal x. That is,

$$\hat{y}_x(t) = \sum_{\tau=1}^{k} h(\tau)x(t-\tau), \quad t = 2,\dots,N+1.$$

Note that $\hat{y}_x$ is a linear function of x.

The nonnegativity and monotonicity constraint on x can be expressed as a set of linear inequalities,

$$x(1) \geq 0, \quad x(1) \leq x(2), \quad \dots \quad x(N-1) \leq x(N).$$

Now we turn to the maximum likelihood problem. The likelihood function is

$$\prod_{t=2}^{N+1} p\left(y(t) - \hat{y}_x(t)\right),$$

where p is the density function of a $\mathcal{N}(0,1)$ random variable. The negative log-likelihood function has the form

$$\alpha + \beta\|\hat{y}_x - y\|_2^2,$$

where α is a constant, and β is a positive constant. Thus, the ML estimate is found by minimizing the quadratic objective $\|\hat{y}_x - y\|_2^2$. The ML estimate when we *do not* take into account signal nonnegativity and monotonicity can be found by solving a least-squares problem,

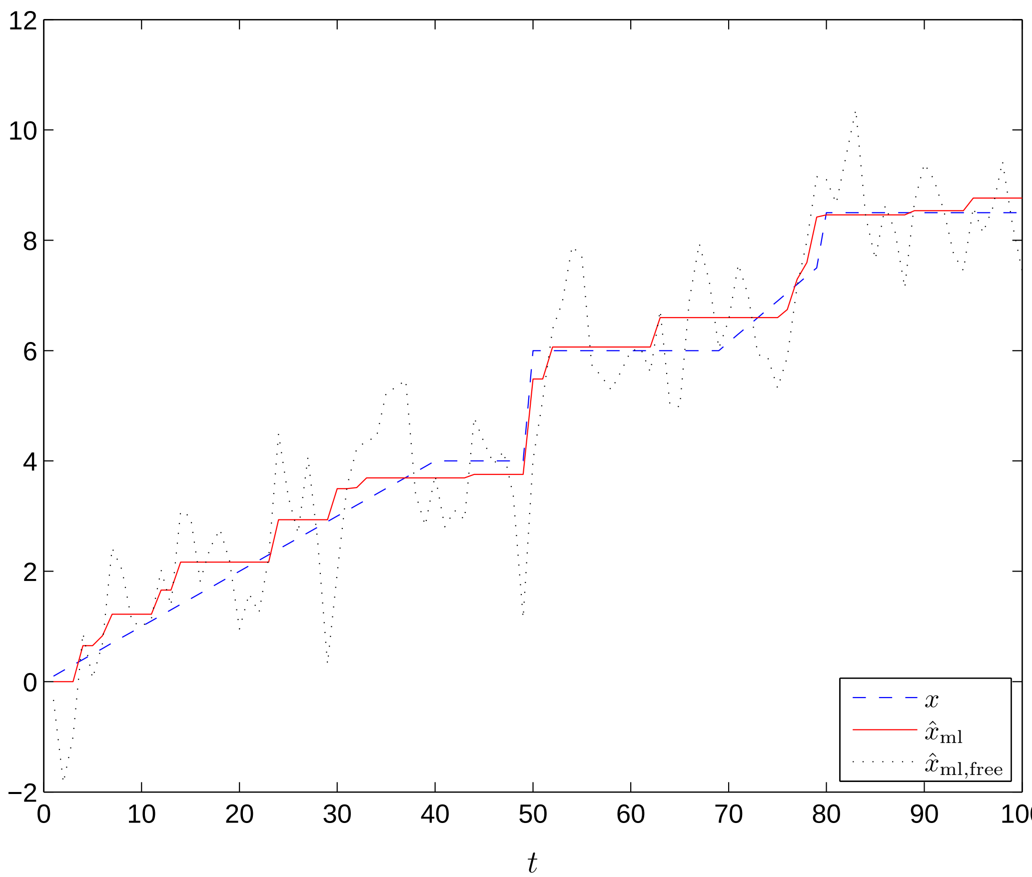
$$\hat{x}_{\rm ml,free} = \operatorname*{argmin}_x \|\hat{y}_x - y\|_2^2.$$

Since convolution is an invertible linear operation, to get the ML estimate without the monotonicity constraint, we simply apply deconvolution. Since the number of measurements and variables to estimate are the same (N), there is no smoothing effect to reduce noise, and we can expect the deconvolved estimate to be poor.

With the prior assumption that the signal x is nonnegative and monotonically nondecreasing we can find the ML estimate $\hat{x}_{\mathrm{ml}}$ by solving the QP

$$\begin{array}{ll} \text{minimize} & \|\hat{y}_x - y\|_2^2 \\ \text{subject to} & x(1) \geq 0 \\ & x(t) \leq x(t+1) \quad t = 1, \ldots, N-1. \end{array}$$

(b) The ML estimate for the given problem instance, with and without the assumption of nonnegativity and monotonicity, is plotted below. We've also plotted the true signal x. We observe that the ML estimate $\hat{x}_{\mathrm{ml}}$, which takes into consideration nonnegativity and monotonicity, is a much better estimate of signal x than the simple unconstrained solution $\hat{x}_{\mathrm{ml,free}}$ obtained via deconvolution.



The following Matlab code computes the ML solutions for the constrained and unconstrained estimation problems.

```
% ML estimation of increasing nonnegative signal
% problem data
ml_estim_incr_signal_data;

% maximum likelihood estimation with no monotonicity taken in to account
% can be solved analytically
cvx_begin
    variable xls(N)
    yhat = conv(h,xls);    % estimated output
    % yhat is truncated to match problem description
    minimize (sum_square(yhat(1:end-3) - y))
cvx_end

% monotonic and non-negative signal estimation
cvx_begin
    variable xmono(N)
    yhat = conv(h,xmono);  % estimated output
    minimize (sum_square(yhat(1:end-3) - y))
    subject to
        xmono(1) >= 0;
        xmono(1:N-1) <= xmono(2:N);
cvx_end

t = 1:N;
figure; set(gca, 'FontSize',12);
plot(t,xtrue,'--',t,xmono,'r',t,xls,'k:');
xlabel('t'); legend('xt','xmono','xls','Location','SouthEast');
%print -depsc ml_estim_incr_signal_plot
```

A6.10 *Worst-case probability of loss.* Two investments are made, with random returns R_1 and R_2. The total return for the two investments is $R_1 + R_2$, and the probability of a loss (including breaking even, *i.e.*, $R_1 + R_2 = 0$) is $p^{\text{loss}} = \mathbf{prob}(R_1 + R_2 \leq 0)$. The goal is to find the worst-case (*i.e.*, maximum possible) value of p^{loss}, consistent with the following information. Both R_1 and R_2 have Gaussian marginal distributions, with known means μ_1 and μ_2 and known standard deviations σ_1 and σ_2. In addition, it is known that R_1 and R_2 are correlated with correlation coefficient ρ, *i.e.*,

$$\mathbf{E}(R_1 - \mu_1)(R_2 - \mu_2) = \rho\sigma_1\sigma_2.$$

Your job is to find the worst-case p^{loss} over any joint distribution of R_1 and R_2 consistent with the given marginals and correlation coefficient.

We will consider the specific case with data

$$\mu_1 = 8, \quad \mu_2 = 20, \quad \sigma_1 = 6, \quad \sigma_2 = 17.5, \quad \rho = -0.25.$$

We can compare the results to the case when R_1 and R_2 are jointly Gaussian. In this case we have

$$R_1 + R_2 \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2),$$

which for the data given above gives $p^{\rm loss} = 0.050$. Your job is to see how much larger $p^{\rm loss}$ can possibly be.

This is an infinite-dimensional optimization problem, since you must maximize $p^{\rm loss}$ over an infinite-dimensional set of joint distributions. To (approximately) solve it, we discretize the values that R_1 and R_2 can take on, to $n = 100$ values $r_1, \ldots, r_n$, uniformly spaced from $r_1 = -30$ to $r_n = +70$. We use the discretized marginals $p^{(1)}$ and $p^{(2)}$ for R_1 and R_2, given by

$$p_i^{(k)} = \mathbf{prob}(R_k = r_i) = \frac{\exp\left(-(r_i - \mu_k)^2/(2\sigma_k^2)\right)}{\sum_{j=1}^n \exp\left(-(r_j - \mu_k)^2/(2\sigma_k^2)\right)},$$

for $k = 1, 2$, $i = 1, \ldots, n$.

Formulate the (discretized) problem as a convex optimization problem, and solve it. Report the maximum value of $p^{\rm loss}$ you find. Plot the joint distribution that yields the maximum value of $p^{\rm loss}$ using the Matlab commands `mesh` and `contour`.

Remark. You might be surprised at both the maximum value of $p^{\rm loss}$, and the joint distribution that achieves it.

Solution. Let $P \in \mathbf{R}_+^{n\times n}$ be the matrix of joint probabilities, with $P_{ij} = \mathbf{prob}(R_1 = r_i,\ R_2 = r_j)$. The condition that the marginals are the given ones is

$$P\mathbf{1} = p^{(1)}, \qquad P^T\mathbf{1} = p^{(2)}.$$

The correlation constraint can be expressed as

$$(r - \mu_1\mathbf{1})^T P (r - \mu_2\mathbf{1}) = \rho\sigma_1\sigma_2.$$

The probability of a loss $R_1 + R_2 \le 0$ is given by

$$\mathbf{prob}(R_1 + R_2 \le 0) = \sum_{r_i + r_j \le 0} P_{ij}.$$

So the problem is an LP,

$$\begin{array}{ll}
\mbox{maximize} & \sum_{r_i + r_j \le 0} P_{ij} \\
\mbox{subject to} & P_{ij} \ge 0, \quad i, j = 1, \ldots, n \\
& P\mathbf{1} = p^{(1)} \\
& P^T\mathbf{1} = p^{(2)} \\
& (r - \mu_1\mathbf{1})^T P (r - \mu_2\mathbf{1}) = \rho\sigma_1\sigma_2,
\end{array}$$

with variable $P \in \mathbf{R}_+^{n\times n}$.

The code to find the worst-case joint distribution is below.

```
% loss bounds solution
clear all; close all;

mu1 = 8; mu2 = 20;
sigma1 = 6; sigma2 = 17.5;
rho = -0.25;

% assuming jointly gaussian distribution
mu = mu1 + mu2;
sigma = sqrt(sigma1^2 + sigma2^2 + 2*rho*sigma1*sigma2);
ploss = normcdf(0, mu, sigma);  % gaussian probability of loss

n = 100;
rmin = -30; rmax = 70;
% discretize outcomes of R1 and R2
r = linspace(rmin,rmax,n)';

% marginal distributions
p1 = exp(-(r-mu1).^2/(2*sigma1^2)); p1 = p1/sum(p1);
p2 = exp(-(r-mu2).^2/(2*sigma2^2)); p2 = p2/sum(p2);

% form mask of region where R1 + R2 <= 0
r1p = r*ones(1,n); r2p = ones(n,1)*r';
loss_mask = (r1p + r2p <= 0)';

cvx_begin
    variable P(n,n)
    maximize (sum(sum(P(loss_mask))))
    subject to
        P >= 0;
        sum(P ,2) == p1;
        sum(P',2) == p2;
        (r - mu1)'*P*(r - mu2) == rho*sigma1*sigma2;
cvx_end

pmax = cvx_optval; % worst case probability of loss
pmax
ploss

P = full(P);
figure(1); mesh(r1p, r2p, P');
xlabel('R1'); ylabel('R2'); zlabel('density');
```

```
xlim([rmin rmax]); ylim([rmin rmax]);
print -depsc 'loss_bounds_mesh.eps'

figure(2); contour(r1p, r2p, P');
xlabel('R1'); ylabel('R2'); grid on;
xlim([rmin rmax]); ylim([rmin rmax]);
print -depsc 'loss_bounds_cont.eps'
```

This yields a probability of loss of approximately 0.192, almost four times larger than the loss probability when R_1 and R_2 are jointly Gaussian! The resulting (worst-case) distribution is plotted below. It has mass on the line where $R_1 + R_2 = 0$ (*i.e.*, break even, which counts as a loss for us). Then it has extra mass on another region in the plane, which is needed to make the marginals the given Gaussians, as well as to meet the constraint on the correlation coefficient.

By the way, we were not picky about the numerics when grading, and gave generous partial credit so long as you grasped the main idea.

Remark. We didn't ask you to show this, but here's the analysis when R_1 and R_2 are jointly Gaussian. Since we are given the means, marginal variances, and correlation, we have

$$(R_1, R_2) \sim \mathcal{N}\left((\mu_1, \mu_2),\ \Sigma\right), \qquad \Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}.$$

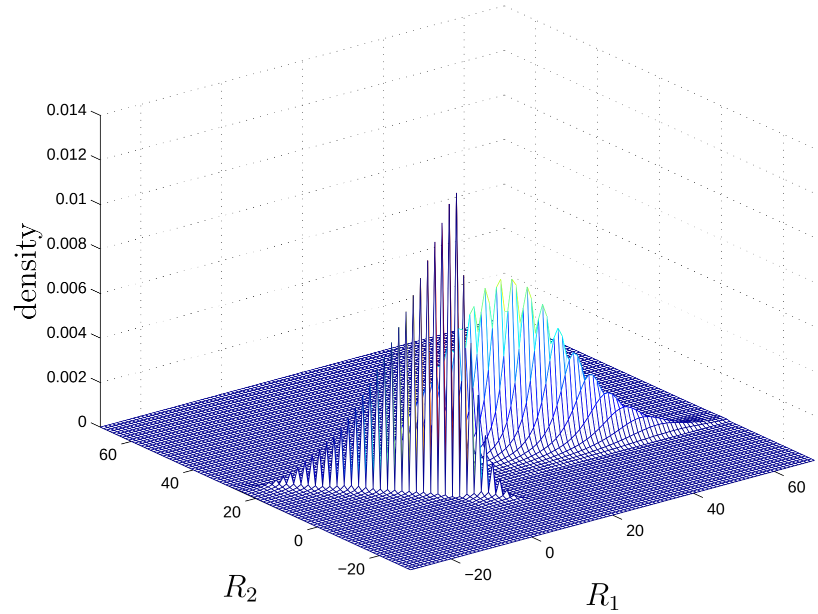
The total return is $R_1 + R_2$ is also Gaussian, with $\mathbf{E}(R_1 + R_2) = \mu_1 + \mu_2$ and variance

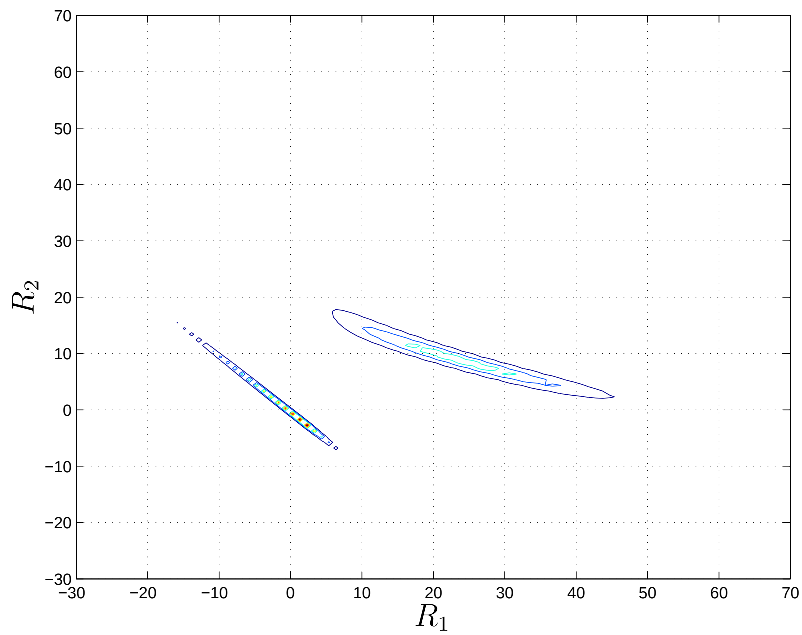
$$\mathbf{var}(R_1 + R_2) = \mathbf{1}^T \Sigma \mathbf{1} = \sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2.$$

The probability of loss for the jointly Gaussian case is therefore

$$p^{\text{loss}} = \Phi\left(-\frac{\mu_1 + \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2}}\right),$$

where $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt$ is the cumulative distribution of a standard Gaussian.


density
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
60
40
20
0
−20
R_2
−20
0
20
40
60
R_1


70
60
50
40
30
20
10
0
−10
−20
−30
R_2
−30
−20
−10
0
10
20
30
40
50
60
70
R_1