Assignment 3 Key for JDEP 384H

Thomas Shores
Department of Mathematics
University of Nebraska
Spring 2007

Points: 45
Due: March 20

1. (8 pts) You are designing a portfolio consisting of the three assets described in Exercise 1 of Assignment 2.

(a) Express the problem of finding the minimum risk portfolio as a quadratric programming problem and use the MatlabTools program `quad_prog.m` to find the minimum risk. (Grab the latest copy of this program from our home directory in MatlabTools for this exercise.)

(b) Use `quad_prog.m` to make a graph of the mean-variance efficient frontier of this portfolio using steps of 0.2% from minimum to maximum return.

Solution.

(a) The minimum risk portfolio with weightings $\mathbf{w} = (w_1, w_2, w_3)$ will solve the quadratic programming program

$$\begin{gathered}\text{Min } Z = \mathbf{w}^T A \mathbf{w} \\ \text{Sbj} \\ w_1 + w_2 + w_3 = 1 \\ w_i \geq 0,\ i = 1, 2, 3.\end{gathered}$$

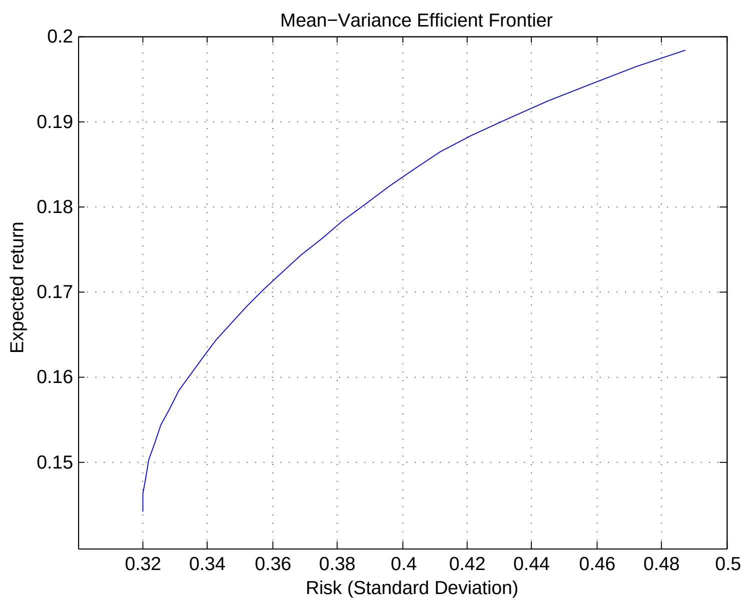
Here $A = DRD$ is the covariance matrix which was calculated from the diagonal of standard deviations and R the correlation matrix as in Exercise 1 of Assignment 2. It is displayed in the following diary file in Matlab:

```
% Part (a)
mystartup % point to Tools directory
D = diag([0.28,0.24,0.25]) % standard deviation diagonal
D =
0.28000 0.00000 0.00000
0.00000 0.24000 0.00000
0.00000 0.00000 0.25000
R = [1 -0.1 0.25; -0.1 1 0.2; 0.25 0.2 1] % correlation matrix
R =
1.00000 -0.10000 0.25000
-0.10000 1.00000 0.20000
0.25000 0.20000 1.00000
S = D*R*D % covariance matrix
S =
0.0784000 -0.0067200 0.0175000
-0.0067200 0.0576000 0.0120000
0.0175000 0.0120000 0.0625000
c = [0;0;0]; % linear term in Z
A = [1, 1, 1;-1, -1, -1]; % account for equality
b = [1;-1]; % rhs for equality
```

```
[wts,zmin] = quad_prog(S,c,A,b) % minimize risk
wts =
0.31597
0.43909
0.24495
zmin = 0.026108
riskmin = sqrt(zmin) % minimum risk
riskmin = 0.16158
```

(b) Here is a diary file of the Matlab used to solve the problem, followed by the graph, which was massaged a bit with Matlab tools.

```
% Part (b)
mu = [0.1;0.15;0.2]; % mean return vector
returnmin = wts'*mu; % return corresponding to min risk
returnmin = wts'*mu % return corresponding to min risk
returnmin =
0.1444
returnmax = max(mu); % maximum risk
returnmax = max(mu) % maximum risk
returnmax =
0.2000
returns = returnmin:0.002:returnmax; % range of returns
risks = zeros(size(returns));
for k = 1:length(returns) % minimize risk at each return level
[wts,zmin] = quad_prog(S,c,[A;mu';-mu'],[b;returns(k);-returns(k)]);
risks(k) = sqrt(zmin);
end
plot(risks,returns)
quit
```



2. (8 pts) You are selling the portfolio of Exercise 1 to a client who is leery of asset 3 and does not want more than 40% of the portfolio in that asset.

(a) Graph an efficient frontier for possible portfolios for this client.

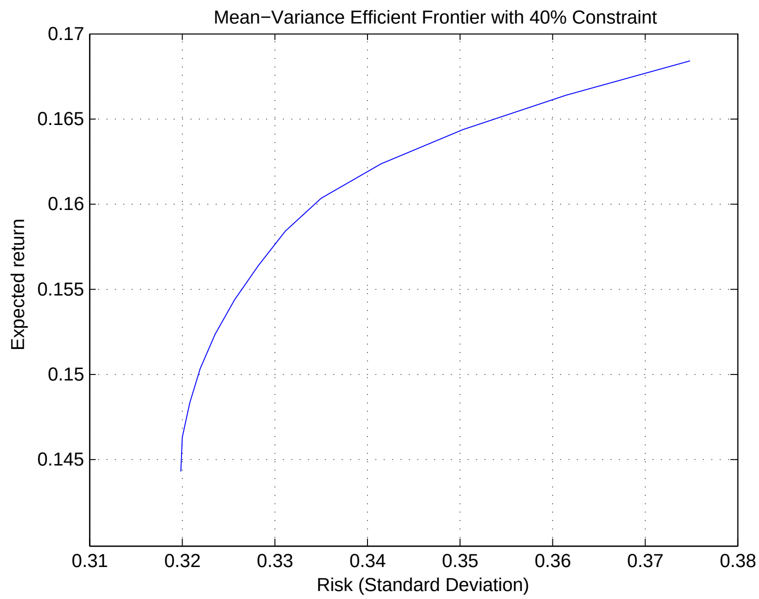
(b) Your client has a target of 17% expected return with, of course, minimum risk. Your advice?

Solution.

(a) The only change to the quadratic programming problem is to add an additional constraint of the form $w_3 \leq 0.4$. However, we need to use a different maximum rate, because we cannot use the obvious weighting $(0, 0, 1)$ to give the maximum return. So we put as much as we can into asset 3 and the rest in the next largest returner, which is asset 2. This gives a weighting of $(0, 0.6, 0.4)$ for the maximum return. Now rework the problem:

```
% Part (a)
mystartup % point to Tools directory
D = sqrt(diag([0.28,0.24,0.25])); % standard deviation diagonal
R = [1 -0.1 0.25; -0.1 1 0.2; 0.25 0.2 1]; % correlation matrix
S = D*R*D; % covariance matrix
c = [0;0;0]; % linear term in Z
A = [1, 1, 1;-1, -1, -1;0, 0, -1]; % account for sum to 1, -w_1>=-0.4
b = [1;-1;-0.4]; % rhs for equality and inequality
[wts,zmin] = quad_prog(S,c,A,b); % minimize risk
riskmin = sqrt(zmin); % minimum risk
mu = [0.1;0.15;0.2]; % mean return vector
returnmin = wts'*mu; % corresponding return
returnmax = [0, 0.6, 0.4]*mu % the obvious maximum return
returnmax =
0.17
riskmax = sqrt([0,0.6,0.4]*S*[0,0.6,0.4]') % corresponding risk
riskmax =
0.3872
returns = returnmin:0.002:returnmax; % range of returns
risks = zeros(size(returns)); % range of risks
for k = 1:length(returns) % minimize risk at each return level
[wts,zmin] = quad_prog(S,c,[A;mu';-mu'],[b;returns(k);-returns(k)]);
risks(k) = sqrt(zmin);
end
plot(risks,returns),grid
xlabel('Risk (Standard Deviation)')
ylabel('Expected return')
title('Mean-Variance Efficient Frontier with 40% Constraint')
quit
```

The efficient frontier that results is:



(b) First, we need to tell the client that while it is mathematically possible to attain that return, it will be the riskiest possible portfolio under those assumptions with a risk of 0.3872. Secondly, at that risk level a close inspection of our mean-variance efficient frontier in the previous Execise indicates that we could achieve a return of nearly 18% at the same risk level. One could also calculate the risk minimization at a level of 17% return and obtain a risk of 0.3565 with a weighting $(0.12, 0.36, 0.52)$

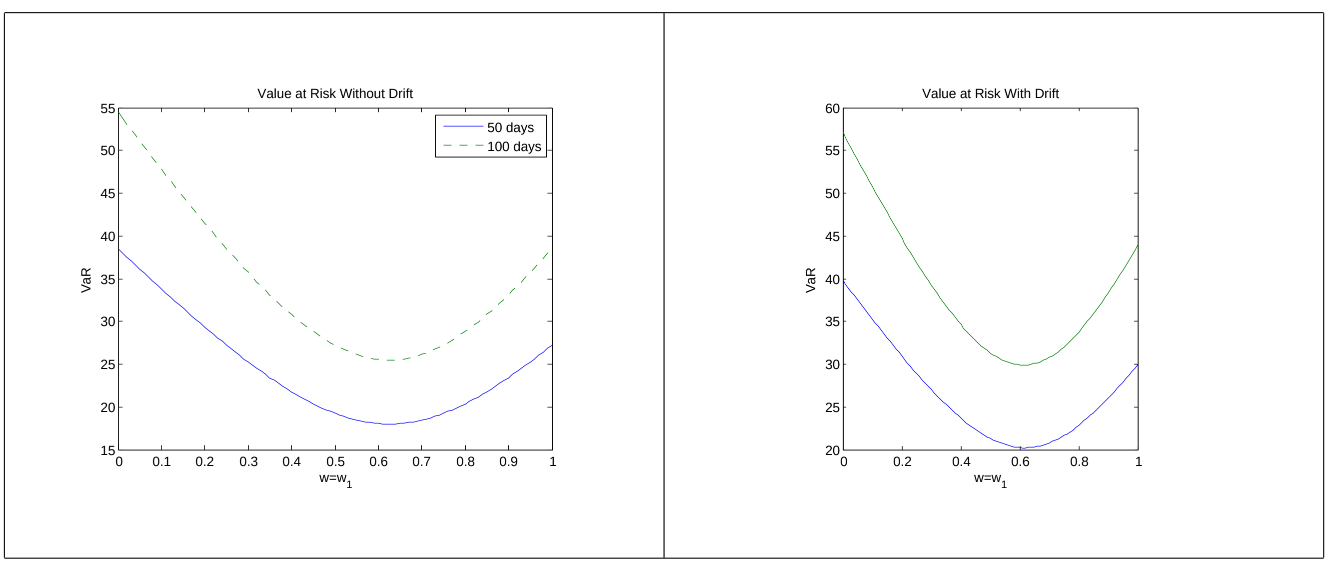
3. (8 pts) Refer to the example portfolio of two assets whose efficient frontier we graphed in Lecture 11. Since there are only two weights, the value of one weight, say $w = w_1$, uniquely determines the portfolio since $w_2 = 1 - w_1$. Make a plot on a single graph of the value at risk of the portfolio as a function of w for time horizons of 50 and 100 days. Assume the variances and in the example are annual, so prorate them for daily rates. Also assume that the value of the portfolio in all cases is \$100.

Solution. The portfolio in question from text, p. 74, has two assets with expected earning rates $\bar{r}_1 = 0.2$, $\bar{r}_2 = 0.1$, $\sigma_1^2 = 0.2$, $\sigma_2^2 = 0.4$ and $\sigma_{12} = -0.1$. No mention of drift was made in the statement of the exercise, so we shall graph the VaR both with and without drift taken into account. We can make the substitution $\mathbf{w} = (w, 1-w)$ and obtain that the variance over the fraction of a year δt is given by

$$\begin{aligned}\sigma^2 &= [w, 1-w]\begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.4 \end{bmatrix}\begin{bmatrix} w \\ 1-w \end{bmatrix}\delta t \\ &= \left(0.8w^2 - w + 0.4\right)\delta t.\end{aligned}$$

Since the confidence level α was not specified, we will take α to be 95% for purposes of our graph. Also, the drift will be the expected (annual) rate prorated for time δt. Here is a diary transcript of the graph generation, followed by the graphs

```
mystartup % point to Tools directory
alpha = 0.95; % confidence level
S = [0.2, -0.1; -0.1, 0.4] % covariance matrix
S =
0.2000 -0.1000
-0.1000 0.4000
mu = [0.2; 0.1] % annual expected return (drift)
mu =
0.2000
0.1000
zalpha = stdn_inv(1-alpha) % quantile
zalpha =
-1.6449
dt1 = 50/365 % first deltat
dt1 =
0.1370
dt2 = 100/365 % second deltat
dt2 =
0.2740
w = (0:.01:1)'; % range of w values
VaR1 = 100*(-zalpha*sqrt((0.8*w.^2-w+0.4)*dt1)); % values at risk for 50 days
VaR2 = 100*(-zalpha*sqrt((0.8*w.^2-w+0.4)*dt2)); % values at risk for 100 days
figure(1)
plot(w,[VaR1,VaR2])
% next, account for drift
VaR1drift = 100*([w,1-w]*mu*dt1-zalpha*sqrt((0.8*w.^2-w+0.4)*dt1)); % values at risk
for 50 days
VaR2drift = 100*([w,1-w]*mu*dt2-zalpha*sqrt((0.8*w.^2-w+0.4)*dt2)); % values at risk
for 100 days
figure(2)
plot(w,[VaR1drift,VaR2drift])
quit
```



4. (5 pts) Give a careful explanation why Ito's Lemma applied to $f(S) = \log S$ with random process $\dfrac{dS}{S} = \sigma\, dW + \mu\, dt$, leads to the formula

$$df = \sigma dX + \left(\mu - \frac{1}{2}\sigma^2\right) dt.$$

How does this compare this to what you get by applying the deterministic chain rule to obtain a differential formula for df?

Solution. Recall that Ito's Lemma says that if $f(S,t)$ is the price of a call or put, and $dS/S = \sigma dX + \mu\, dt$, dX the risky part,then

$$df = \sigma S\frac{\partial f}{\partial S}dX + \left(\mu S\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 f}{\partial S^2} + \frac{\partial f}{\partial t}\right) dt.$$

Now take $f(S,t) = \log S$. In the formula $dS/S = \sigma dW + \mu\, dt$, dW is the risky part, so here $dW = dX$. It follows that $\partial f/\partial t = 0$, $\partial f/\partial S = 1/S$ and $\partial^2 f/\partial S^2 = -1/S^2$. Plug these derivatives into the Ito formula and obtain that

$$\begin{aligned}
df &= \frac{\partial f}{\partial S}\sigma S dX + \left(\frac{\mu S}{S} + \frac{1}{2}\sigma^2 S^2\left(-\frac{1}{S^2}\right)\right) dt \\
&= \frac{S\sigma dX}{S} + \left(\frac{\mu S}{S} + \frac{1}{2}\sigma^2 S^2\left(-\frac{1}{S^2}\right)\right) dt \\
&= \sigma\, dX + \left(\mu - \frac{1}{2}\sigma^2\right) dt,
\end{aligned}$$

as desired.

If we use the deterministic chain rule, then the correct formula i

$$\begin{aligned}
df &= \frac{\partial f}{\partial S}dS + \frac{\partial f}{\partial t}dt \\
&= \frac{dS}{S} + 0 \\
&= \sigma\, dX + \mu\, dt.
\end{aligned}$$

This is nearly the same as Ito's lemma, except that the coefficient of dt is μ instead of $\left(\mu - \frac{1}{2}\sigma^2\right)$, which the stochasticity of S introduces.

5. (8 pts) Consider the random walk suggested by the differential equation $dS = \sigma S dX + \mu S dt$ with $\mu = 0.1$ and $\sigma = 0.3$ and time in units of years.

(a) Create a graph with 10 simulations of this random walk, starting with $S(0) = 100$, and in steps of $dt = 1/12$ over a two year period. Before you begin, reset the random number generator of Matlab with the command `randn('state',0)`.

(b) Calculate $S(2)$ by running 100 simulations of this walk and compute the mean and variance of this sample. How does the sample mean and variance compare with the formula at the bottom of page 99 of the text and the volatility, respectively?

Solution.

(a) We use the straightforward discretization of differentials given by

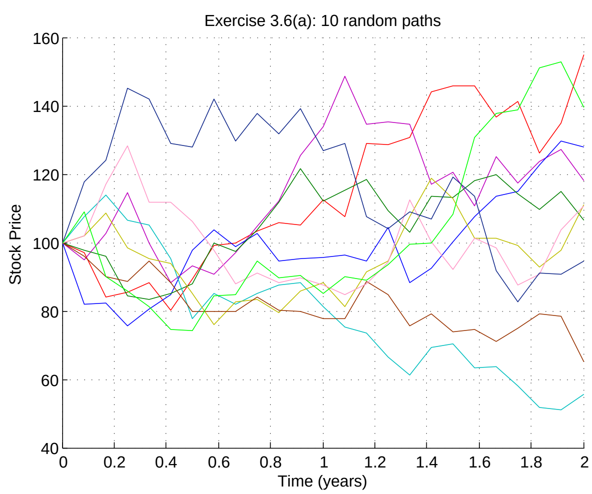
$$\Delta S = S(t + \delta t) - S(t) \approx \mu S(t)\ \delta t + \sigma S(t)\ dX$$

where $dX = \sqrt{\delta t} Z$ with standard normal distribution $Z \sim N(0,1)$ had variance δt and mean zero. This translates into

$$S_{k+1} = S_k \left(\mu\, \delta t + \sigma \sqrt{\delta t} z\right)$$

for some random sample z of Z. Here is a script that generates the graph, followed by the graph:

```
% script:  Exercise3_5a.m
mu = 0.1;
sigma = 0.3;
dt = 1/12;
xnodes = linspace(0,2,25);
S = zeros(25,1);
S(1) = 100;
figure
hold on, grid
randn('state',0)
for j = 1:10
for k = 1:24
S(k+1) = S(k)*(1 + mu*dt + sigma*sqrt(dt)*randn());
end
plot(xnodes,S)
end
```



(b) Following is the script that generates the answer:

```
% script:  Exercise3_5b.m
mu = 0.1
sigma = 0.3
dt = 1/12
T = 2;
S = zeros(25,1);
S2 = zeros(100,1);
```

```
S(1) = 100;
randn('state',0)
for j = 1:100
for k = 1:24
S(k+1) = S(k)*(1 + mu*dt + sigma*sqrt(dt)*randn());
end
S2(j) = S(25); % store up results at 2 years
end
truemean = S(1)*exp(mu*2) % according to page 99
samplemean = mean(S2)
samplevar = var(S2)
% just for the record
truevar = exp(2*(log(S(1))+(mu-sigma^2/2)*T) + sigma*sqrt(T))*(exp(sigma^2*T)-1)
```

This script generates the following output:

```
Exercise3_5b
truemean =
122.1403
samplemean =
126.1687
samplevar =
3.2861e+003
truevar =
3.7562e+003
diary off
```

We can see from the output that the sample mean of 126.1 and true mean of 122.1 compare favorably. However, the sample variance 3286 is very far from the volatility 0.3

(Just for the record, one can use the formula on page 632 of the text to obtain that the true variance is actually 3756, to which the sample variance is fairly comparable.)

6. (8 pts) Consider a call option with strike price of 50 on a stock at a time that is 5 months before expiry. Assume that the volatility of this stock is $\sigma = 0.4$ and that the risk-free interest rate is 10%.

(a) Make a graph of the payoff curve and the price of the option for S ranging from 20 to 100, assuming that the call is European (use `bseurcall.m`.)

(b) On the same graph as (a) plot the price of the option assuming that it is American (use `LatticeAmCall.m`, lattice of 100 time steps, and a for loop.) Comment on the plots.

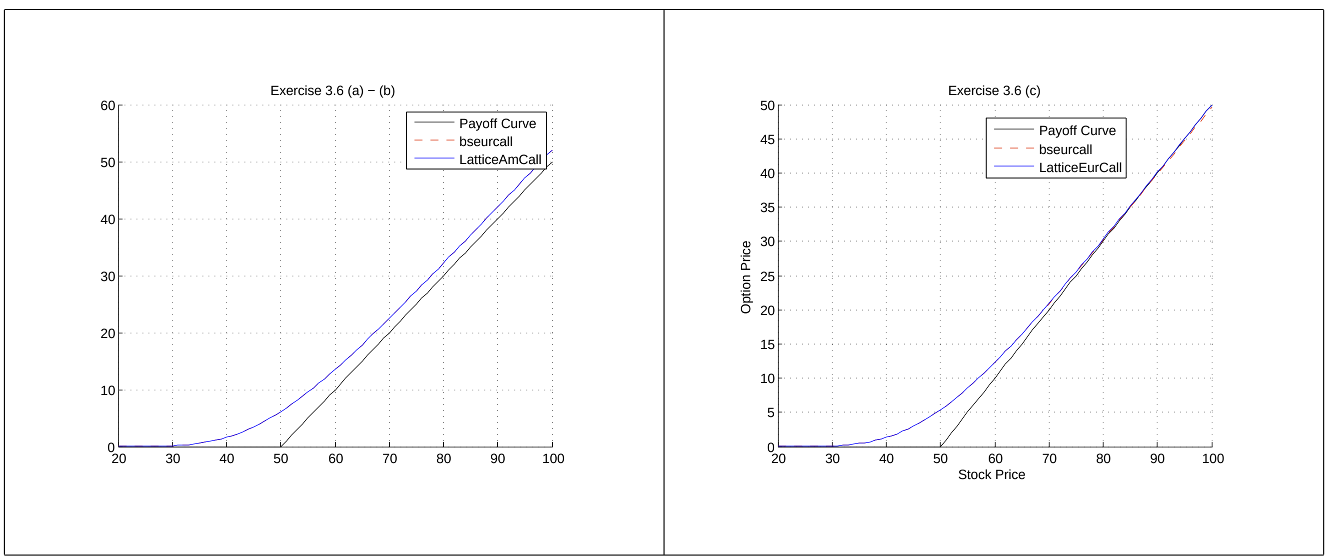
(c) Repeat (a) and (b) under the assumption that the stock pays dividends at a continuous rate of $D_0 = 0.06$.

Solution.

(a-c) Here is the script that generated the graphs of (a)-(b) and (c), followed by the graphs.

```
% script:  Exercise3_6abc.m
mystartup % make right directories available
t = 0 % current time
K = 50 % strike price
```

```
r = 0.1 % current risk-free interest rate
T = 5/12 % expiry
sigma = 0.4 % volatility
S = 20:1:100; % interval of interest
D0 = 0 % no dividends
EC = bseurcall(S,K,r,T,t,sigma,D0);
payoff = max(S-K,0); % payoff curve
% now build the American call, term by term
N = 100; AC = S;
for ii = 1:length(S)
AC(ii) = LatticeAmCall(S(ii),K,r,T,sigma,N,D0);
end
figure(1) % figure for parts (a) and (b)
hold on, grid % so we can overlay graphs and discriminate
plot(S,payoff) % plot payoff curve
plot(S,EC) % plot European call
plot(S,AC) % plot American call
D0 = 0.06; % calculations for part (c)
EC = bseurcall(S,K,r,T,t,sigma,D0);
payoff = max(S-K,0); % payoff curve
% now build the American call, term by term
N = 100; AC = S;
for ii = 1:length(S)
AC(ii) = LatticeAmCall(S(ii),K,r,T,sigma,N,D0);
end
figure(2) % figure for part (c)
hold on, grid % so we can overlay graphs and discriminate
plot(S,payoff) % plot payoff curve
plot(S,EC) % plot European call
plot(S,AC) % plot American call
```



Notice that the plots of American and European calls are identical when $D_0 = 0$ as in (a) and (b), but that in (c), where there is a positive dividend, the American option stays above the payoff curve, but the European option falls below the payoff curve.