Assignment 5 Key for JDEP 384H

Thomas Shores
Department of Mathematics
University of Nebraska
Spring 2007

Points: 45
Due: April 25

1. (10 pts) For this exercise, use the data you can load from the file AsianExampleData5_1 which can be found in the directory Week15. You are to price an Asian call with the parameters in this file. Follow the Asian call example in Lecture 24.

(a) Identify the approximation to the price and corresponding confidence interval of the option with these parameters, using the simple Monte Carlo method of Lecture 24.

(b) Use the price of a European call with the same parameters as a control variate and calculate the approximation to price and confidence interval. Is there any improvement over (a)?

Solution.

(a) Here is the Matlab diary file for (a)

```
% Exercise 5.1(a)
AsianExampleData5_1
alpha =
0.0500
sigma =
0.3000
S0 =
100
r =
0.0600
K =
110
mu =
0.1000
nsteps =
50
T =
0.5000
dt =
0.0100
nreps =
1000
Spaths = AssetPath(S0,mu,sigma,T,nsteps,nreps);
SFinal = mean(Spaths(:,2:nsteps+1)')';
prices = exp(-r*T)*max(0,SFinal - K);
% Results with no control variate:
[price, V, CI] = norm_fit(prices)
price =
2.4578
V =
```

```
5.6145
CI =
2.1098
2.8058
```

(b) Here is the rest of the Matlab diary:

```
Cn = max(0,Spaths(:,nsteps+1)); % control variate
muC = mean(Cn); % expected value of C
S = cov([prices,Cn]); % calculate covariance matrix
bta = -S(2,1)/S(2,2) % estimate bta
bta =
-0.1748
pricesC = prices + bta*(Cn - muC);
% Results with control variate:
[pricesC, V, CI] = norm_fit(pricesC)
pricesC =
2.4578
V =
4.0446
CI =
2.2071
2.7084
```

There is modest improvement with the control variates insofar as the variance of the sample was reduced from `5.6145` to`4.0446` which in turn reduced the size of the confidence interval from`[2.1098, 2.8058]` to`[2.2071,2.7084]`. Interestingly, though, the computed mean remained the same.

2. (10 pts) Modify the down-and-out put example of Lecture 24 to approximate the value of an up and out call with the same data as in AsianExampleData5_1.m and an up barrier of $S_b = 125$. Find an estimate of the lowest value of S_b that puts the value of this option within 90% of the value of the corresponding call without a barrier.

Solution.

By trial and error (and resetting the random number generator each time to obtain a fair comparison), we were able to determine that $Sb = 160$ was about as low as we could go and keep the Monte Carlo estimate above 90% of the corresponding European option call price.

However, if we tried to keep the entire confidence interval in that range it was a different story, with $Sb = 170$ being the minimum the barrier could be and be within 90% of the corresponding European option call price. Here is the Matlab diary file that we used:

```
AsianExampleData5_1
alpha =
0.0500
sigma =
0.3000
S0 =
100
r =
0.0600
K =
```

```
110
mu =
0.1000
nsteps =
50
T =
0.5000
dt =
0.0100
nreps =
1000
Sb = 160
Sb =
160
% Price of analogous European call:
% We want the up-and-out to be at least 90% of European call:
minval = 0.9*bseurcall(S0,K,r,T,0,sigma,0)
minval =
5.1898
Spaths = AssetPath(S0,mu,sigma,T,nsteps,nreps);
SFinal = Spaths(:,nsteps+1);
Indicator = (sum((Spaths>Sb)')')==0;
prices = exp(-r*T)*max(0,(SFinal-K).*Indicator);
% Monte Carlo estimate of up-and-out call:
[smplmu,smplstdv,muci] = norm_fit(prices,alpha)
smplmu =
5.2070
smplstdv =
9.5165
muci =
4.6172
5.7968
```

3. (13 pts) Consider the game with payoff table

Strategy		Player 2: 1	2	3	4
Player 1	1	3	-2	1	1
	2	0	2	1	0
	3	1	0	-2	2

(a) Does dominated strategy elimination or minimax/maximin yield a solution to the game?

(b) Solve this game with Matlab. (It would be more elegant to write a program with the syntax `[x,y,p] = gamesolve(A)`, where A is the payoff table, x, y, p the optimal strategies and payoff, but this is not required.)

(c) Is this a fair game? If not, what shift in payoffs would make it fair?

Solution.

(a) No one move for either player is dominated by another of the players moves. Consequently, dominated strategies elimination fails to resolve the game.

The maximin strategy leads player 1 to choose move 2 with a resulting payoff of 0. It leads player 2 to choose move 3 with a resulting payoff of 1. These do not agree, so pure strategies will lead to instability.

(b) The following function solves the game:

```
function [xopt,yopt,payoff] = ZeroSumGameSolve(A)
% usage:  [xopt,yopt,payoff] = ZeroSumGameSolve(A)
% description:  This code solves a two-person zero-sum
% game with payoff matrix A for player 1.  It returns
% player 1 optimal strategy in xopt, and optionally,
% player 2 optimal strategy in yopt and payoff of
% the game in payoff.
% set up for a linprog application
[m,n] = size(A);
Amin = min(min(A));
Amin = Amin*(Amin<0);
A1 = A - Amin;
% inequality A1*x<=b1 is required, so...
A1 = [-A1',ones(n,1)];
b1 = zeros(n,1);
% now the equality
E = [ones(1,m),0];
e = 1;
% finally, the objective function
c = -[zeros(m,1);1];
xopt = linprog(c,A1,b1,[],[],E,e);
payoff = xopt(m+1) + Amin;
xopt = xopt(1:m);
% next the =dual problem
% inequality A1*y<=b1
A1 = A - Amin;
A1 = [A1,-ones(m,1)];
b1 = zeros(m,1);
% now the equality
E = [ones(1,n),0];
e = 1;
% finally, the objective function
c = [zeros(n,1);1];
yopt = linprog(c,A1,b1,[],[],E,e);
yopt = yopt(1:n);
```

Here is a transcript of the computed output which yields the optimal strategy x for player 1, y for player 2 and the resulting payoff:

```
A = [3 -2 1 1;0 2 1 0;1 0 -2 2]
A =
3 -2 1 1
0 2 1 0
```

```
1 0 -2 2
[x,y,payoff]=ZeroSumGameSolve(A)
x =
0.28571
0.57143
0.14286
y =
0.00000
0.14286
0.28571
0.57143
payoff = 0.57143
```

(c) No, it is not fair since the payoff, $4/7 \approx 0.57143$, is not zero. It could be made fair if the payoffs were all decreased by the payoff $4/7$, since the new payoff would be zero.

4. (12 pts) A credit manager in the small business division NMO bank approves (or rejects) credit lines for small businesses. A customer is requesting a credit line of \$100,000 for her specialties clothing store. Experience shows that 20% of customers in this category are poor risks, 50% are average and the rest are good risks. If credit is extended, the average profit is -15% for poor risks, 10% for average risks and 20% for good risks. For \$4,500 the credit manager could obtain additional information from a credit-rating company whose track record with the bank is as follows: If a customer turns out to be a poor risk, the company rates him as poor 45% of the time and average in 35% of the time. If the customer is an average risk, the company rates him average 60% of the time and poor 35% of the time. If the customer is a good risk, the company will rate him as good 50% of the time and average 30% of the time.

(a) Write a decision analysis form for this problem by identifying the states of nature, alternatives, prior probabilities, payoff table and conditional probability table of the credit-rating firm.

(b) Use Bayes' decision rule to determine which course of action should be taken if the credit-rating firm is not used.

(c) Find the EVPI and use it to decide whether or not to seek additional information from the credit-rating firm.

(d) Compute the probabilies that the credit-rating company will rate a customer as a poor, average or good credit risk.

Solution.

(a) The states of nature are that the customer is a poor, average or good risk.

The alternatives are to grant or deny the credit line.

The prior probabilities and payoff table are as follows:

Payoffs	States of Nature (Risk Status)		
Alternatives	Poor	Average	Good
Grant Credit	$-\$15,000$	\$10,000	\$20,000
Deny Credit	\$0	\$0	\$0
Prior Probabilities	0.2	0.5	0.3

The conditional probability table for the credit-rating firm are as follows:

Strategy		Actual (in percent) Poor	Average	Good
Evaluation	Poor	45	35	20
	Average	35	60	30
	Good	20	5	50

(b) Using Baye's decision rule requires us to calculate the expected return of each alternative using prior probabilities. In the case of denying credit, the expected payoff is obviously zero. If credit is granted, the expected payoff is

$$\begin{aligned} E\left[\text{Payoff}|\text{Grant Credit}\right] &= 0.2 \cdot (-15000) + 0.5 \cdot 10000 + 0.3 \cdot 20000 \\ &= 8000. \end{aligned}$$

Therefore, we should choose the alternative of granting credit.

(c) The expected value of perfect information is

$$EVPI = 0.2 \cdot 0 + 0.5 \cdot 10000 + 0.3 \cdot 20000 - 8000 = 3000.$$

Therefore, we should not proceed with the possibility of seeking additional information.

(d) For this, we need only use the law of total probability as in Lecture 27, to deduce that the vector of unconditional probabilities are simply the matrix of conditional probabilities multiplied by the vector of prior probabilities. In Matlab we have

```
Conditionals = [.45 .35 .2;.35 .6 .3; .2 .5 .5]
Conditionals =
0.45000 0.35000 0.20000
0.35000 0.60000 0.30000
0.20000 0.05000 0.50000
priors = [.2; .5; .3]
priors =
0.20000
0.50000
0.30000
unconditionals = Conditionals*priors
unconditionals =
0.32500
0.46000
0.21500
```

Hence the credit-rating firm rates 32.5% of customers a poor risk, 46% of customers an average risk, and 21.5% of customers a good risk.