Mälardalens University
Mathematics and physics department
Project in analytical finance I

västerås 2012-10-22

Valuation of Asian Option

Qi An 910402-T077
Jingjing Guo890103-T107

Content

1. Asian option

Definition of Asian option always emphasizes the gist that the payoff depends on the average price of the underlying asset over a certain period of time as opposed to at maturity. That is why Asian option could also be recognized as average value option.

This type of option contract emerges and gets developed because it tends to cost less than regular American and European options. It is also known that an Asian option can protect an investor from the volatility risk that comes with the market.

There are two basic forms of Asian option, namely Average Price Option and Average Strike Option. The former indicates to pay the difference between the strike and the average of underlying price, and the latter means taking the average price over specified period as the strike of the option and pays the difference between this strike and underlying market price. These could be sorted as follow.

types		payoff
European option	Call	max(S(T)-K,0)
	Put	max(K-S(T),0)
Average price option	Call	max(A(T)-K,0)
	Put	max(K-A(T),0)
Average strike option	Call	max(S(T)-A(T),0)
	Put	max(A(T)-S(T),0)

There are three terms that must be specified.

1) The averaging period. E.g. the last three month, the entire term of option etc.
2) The sampling frequency. E.g. daily, weekly, annually etc.
3) The averaging method.

①simple arithmetic average

$$A_i = \frac{1}{N}\sum_{j=1}^{N} S_i(t_j)$$

②geometric average

$$A(T) = \sqrt[n]{\prod_{i=1}^{n} S_i}$$

③weighted average

i.arithmetic weighted average

$$A(T) = \frac{\sum_{i=1}^{n} w_i S_i}{\sum_{i=1}^{n} w_i}$$

ii.geometric weighted average

$$A(T) = \sqrt[w_1 w_2 \ldots w_n]{S_1^{w_1} S_2^{w_2} \ldots S_n^{w_n}}$$ where w_n denotes the weight.

Basic math knowledge infers that geometric gives more accurate solution. Yet the arithmetic average is more widely used.

2. Pricing

Three main approaches exist to pricing Asian options.

1) European-style options based on geometric averages can be priced by adapting the analytical models. The reason is if underlying price is assumed to follow log normal distribution, then its geometric average is also log normal distributed.
2) The solution above does not apply when it is based on arithmetic averages, because the arithmetic average will not be log normal distributed even if the underlying price is. It is only possible to make approximation.
3) Monte Carlo simulation could be used no matter what its style or averaging method is. As this simulation requires intensive calculation it is mostly performed by computer program.

3. Monte Carlo Simulation

Monte Carlo simulation produces results by constructing a stochastic model, where the aim solution is the expectation of the model, and then use a large number of sample results to approximate the aim solution. The law of large numbers ensures that sample mean converges to the population mean μ almost surely as $n \to \infty$. That is the reason why the sample mean is used as an estimate of population mean for large samples.

Pricing with Monte Carlo method is assumed to be in risk-neutral world. Firstly to generate a path of underlying price stochastically, then compute the expectation of returns, at last discount it with risk-free interest rate.

In this report, Monte Carlo simulation is introduced to value an Asian relative option, which performs on two or more underlying assets, and the payoff is determined by the return of one or more assets relative the return of some other assets.

To make it simpler the common case of two underlying assets is taken.

The price and payoff are given by

$$\prod(0) = e^{-rT} E^Q\left[\prod(T)\right] = e^{-rT} E^Q\left[\max\left\{\sum_{t=1}^{d} \phi_i \frac{A_i}{S_i(t_0)}, 0\right\}\right]$$

where ϕ_i denotes the weight for each return. In this simple case ϕ_1 is 1 and ϕ_2 is -1.

And $S_i(t_j)$ is the price of underlying asset i at time t_j, d is the number of underlying assets.

The Asian sum A_i is defined by $A_i = \frac{1}{N}\sum_{j=1}^{N} S_i(t_j)$, based on simple arithmetic average.

Then it gives us $\prod(0) = e^{-rT} E^{Q}\left[\max\left\{\frac{A_1}{S_1(t_0)} - \frac{A_2}{S_2(t_0)}, 0\right\}\right]$

In our case there are two variables S_1 and S_2 ,which are both under Black-Scholes model with no dividend, and satisfy

$$dS_t = rS_t dt + \sigma S_t dZ_t$$

where r is risk-free interest rate, σ is the standard deviation of return on the asset price, dZ_t is the standard Brownian motion under Q. So the arithmetic average price of Asian option is

$$C = e^{-rT} E_0^Q\left[\max\left\{A(T) - K, 0\right\}\right]$$

Where A(T) is arithmetic mean of price ,K is the strike Price , T is time to maturity, and E is the expectation at t=0 under Q. Because the effective unbiased estimation of E is sample mean, we can rewrite the equation as following,

$$\hat{C}(T,n) = \frac{1}{n} e^{-rT} \sum_{i=1}^{n} \max\left\{\hat{A}_i(T) - K, 0\right\}$$

Where A (T) is the i-th simulated arithmetic mean price, n is the number of simulation times.
Now we can write some scripts on MATLAB to calculate the price of Asian Options by Monte-Carlo simulation.

First is to simulate paths of asset price, and then to calculate the options' price. (As performed in appendix)

Now we have our program for the calculation, then we assume some datum to calculate the price and analyze the properties.

option	Current Price	Strike Price	Interest Rate	Time to Maturity	Volatility
1	\$100	\$110	0.1	1 year	0.4
2	\$100	\$110	0.2	1 year	0.4
3	\$100	\$110	0.1	2 year	0.4
4	\$100	\$110	0.1	1 year	0.2

Then we substitute the datum to the function, we can get the results quickly,

option	Number of Steps	Number of Simulations	Option price	Confidence interval	Confidence	Length of confidence interval
1	10	10	6.6997	(-1.1735,14.5728)	95%	15.7463
	10	10	6.4892	(-4.9774,17.9559)	95%	22.9333
	100	100	7.9192	(5.0215,10.8169)	95%	5.7954

	1000	1000	7.5875	(6.6911,8.4840)	95%	1.7929
2	1000	1000	9.1875	(8.2502,10.1248)	95%	1.8746
3	1000	1000	12.0338	(10.6903,13.3772)	95%	2.6869
4	1000	1000	2.7074	(2.3402,3.0745)	95%	0.7343

It's easy to see that, with the increase of the number of simulations, the length of confidence intervals become smaller and smaller, which means the results of simulations are more and more accurate. So we could obtain an option price very close to its real price.

In option 2, 3 and 4, we changed some values of the parameters so that we have the chance to analyze the effects caused by those parameters roughly. Since we get different prices by using the exactly same values in option 1, we couldn't get some general rule about how the other parameters affect the price by a single result. We need to do more calculations. The reason is every time we use this function in the MATLAB, the system will not create the exactly same random samples. But we can prove we can get more accurate price with Monte-Carlo simulation as the number of random samples close to infinity.

4. Conclusion

Asian Options are commonly traded on currencies and commodity products which have low trading volumes. They are options that ensure to its buyer an average return at a lesser risk. Monte-Carlo simulation is a very useful tool when pricing the Asian option... Also when we try to get some solutions, we need to control all the variables and summarize the general results based on a vast number of trials.

5. References

1 Lecture notes in Analytical Finance 1, Jan R. M. Röman

2 Kijima, M. 2003. Stochastic processes with application to finance, Chapman & Hall/CRC, Boca Raton. ISBN: 1-58488-224-7.

3. John.C.Hull.2007. Options, Futures and Other Derivatives Pearson Education; 6th International edition, ISBN: 0131977059

6. Appendix

Matlab script.

```
function [Price,ConInterval] = aapapricemc (CurrentPrice,StrikePrice,...
    IntRate,TimeToMaturity,StdDeviation,CallOrPut,NumOfSteps,NumOfPaths)
% AAPAPRICEMC    pricing arithmetic average price of Asian option with
% Monte-Carlo Simulation
% PARAMETERS:
% Price is the option price calculated with Monte-Carlo Simulation
% ConInterval is the 95% confidence interval for the estimation of the price
% CurrentPrice is the current price of the underlying asset
% StrikePrice is the strike price of the option
% IntRate is the risk-free interest rate
```

```
% TimeToMaturity is the time to maturity
% StdDeviation is the standard deviation of the return on the price of the underlying asset
% CallOrPut : if the option is a call, CallOrPut is 1;
%             if the option is a put,   CallOrPut is 0;
%             in default , CallOrPut is 1;
if (CallOrPut~=0)&&(CallOrPut~=1)
    error(sprintf('CallOrPut should be 1 or 0.'))
end
Paths = pricepaths(CurrentPrice,IntRate,TimeToMaturity,StdDeviation,...
                   NumOfSteps,NumOfPaths);
if CallOrPut
    PayOff = max(mean(Paths(:,2:(NumOfSteps+1)),2)-StrikePrice,0);
else
    PayOff = max(StrikePrice-mean(Paths(:,2:(NumOfSteps+1)),2),0);
end
[Price,VarOfPrice,ConInterval]=normfit(exp(-IntRate*TimeToMaturity)*PayOff)
end


function Paths = pricepaths ( CurrentPrice , ExpRate , TimeToMaturity ,StdDeviation ,
NumOfSteps , NumOfPaths)
% PricePaths simulating paths of asset price which follows the geometric
% Brownian motion
%
% PARAMETERS:
% Paths is NumOfPaths price paths simulated with NumOfSteps steps;
% CurrentPrice is the current price of the asset;
% ExpRate is the expected rate of return on the asset price;
% TimeToMaturity is the time to maturity;
% StdDeviation is the standard deviation of return on the asset price;

DeltaT = TimeToMaturity/NumOfSteps;
Drift = (ExpRate-StdDeviation^2/2)*DeltaT;
Volatility = StdDeviation*sqrt(DeltaT)*randn(NumOfSteps,NumOfPaths);
Increments = Drift+Volatility;
LogPaths = cumsum([log(CurrentPrice)*ones(NumOfPaths,1),Increments],2);
Paths = exp(LogPaths);
end
```