Linköpings tekniska högskola
IEI
Produktionsekonomi

Finansiell optimering
Laboration 2
November 7, 2018

Computer exercise 2 - Optimal investments - unbounded optimization

Aim: To get experience of using Monte-Carlo simulation and to be able to solve simple stochastic programming problems.

Background: With a scenario based representation of the future general decision problems can be modelled and solved using stochastic programming. When there are no constraints in the optimization problem, then these can be easily solved using methods for unbounded optimization. A common tool on financial markets for developing models is Matlab. You will now implement a solver for unconstrained stochastic programming problems in Matlab.

Download unbounded.zip from lisam which includes Matlab programs for reading historical data, calculating statistics and generating scenarios.

Preparation: Get familiar with the included Matlab programs (begin with runUnbounded.m) and the file sharePrices.xlsx.

Assume that an investor can invest the capital in n different assets which initially cost \$1 and in scenario $i \in \mathcal{L}$ have price $c_i \in \mathbb{R}^{n\times 1}$ with probability, p_i.

The investor can also invest at the continuously compounded risk-free rate, r, which gives the growth $R = e^{rt}$ for time period t. Given the investment $w \in \mathbb{R}^{n\times 1}$ the investor will in scenario i have the total wealth $W_i = c_i^T w + (W - \mathbf{1}^T w)R$, where W is the initial wealth and $\mathbf{1}$ is a column vector of ones. For the initial wealth $W = 1$, w will correspond to the share of wealth invested in the assets. When a power utility function,

$$U(z) = \begin{cases} \frac{z^\gamma}{\gamma} & \gamma \leq 1, \gamma \neq 0 \\ \ln z & \gamma = 0 \end{cases} , \tag{1}$$

is used this will result in the same optimal investment shares, $\alpha = w/W$, independently of the initial wealth. Therefore the problem is solved for $W = 1$.
The optimization problem is then

$$\max_w f = \sum_{i\in\mathcal{L}} p_i U\left(c_i^T w + (1 - \mathbf{1}^T w)R\right). \tag{2}$$

The objective function f can be rewritten as

$$f = \sum_{i\in\mathcal{L}} p_i U\left(c_i^T w + \left(1 - \mathbf{1}^T w\right) R\right) = \sum_{i\in\mathcal{L}} p_i U\left(\left(c_i - \mathbf{1}R\right)^T w + R\right). \tag{3}$$

To solve the unbounded optimization problem with Newtons method, requires the gradient

$$\nabla_w f = \sum_{i \in \mathcal{L}} p_i U' \left((c_i - \mathbf{1}R)^T w + R \right) (c_i - \mathbf{1}R) \tag{4}$$

and the Hessian

$$\nabla_w^2 f = \sum_{i \in \mathcal{L}} p_i U'' \left((c_i - \mathbf{1}R)^T w + R \right) (c_i - \mathbf{1}R) (c_i - \mathbf{1}R)^T \tag{5}$$

where $U'(z) = \frac{dU(z)}{dz}$ and $U''(z) = \frac{d^2U(z)}{dz^2}$.

Preparation: Rehearse Newtons method, and write down the search direction for Newtons method.

. . . .

Use Newtons method without line search, i.e. use the step length $\lambda = 1$. The problem can be solved in principle as unbounded, but it has to be considered that the objective function value is unbounded for the case $0 < \gamma < 1$ when $c_i^T w + (1 - \mathbf{1}^T w)R < 0$, it is also unbounded for the case $\gamma \leq 0$ when $c_i^T w + (1 - \mathbf{1}^T w)R \leq 0$. The step length therefore has to be modified to ensure that the wealth remains positive, i.e. the holdings in the next iteration is updated as $w + \lambda \Delta w$ where

$$\lambda = \min \left\{ \beta \cdot \min_{i \in \mathcal{L}} \left\{ \frac{(c_i - \mathbf{1}R)^T w + R}{-(c_i - \mathbf{1}R)^T \Delta w} \;\middle|\; (c_i - \mathbf{1}R)^T \Delta w < 0 \right\}, 1 \right\}, \tag{6}$$

where $0 < \beta < 1$ is a value, e.g. 0.99, that make certain that the wealth never becomes negative. Note that $\gamma < 1$ has to hold to be able to solve the problem.

Exercise: Implement the solver in unboundedOpt.m, which given the scenarios, can determine the optimal investments. How many iterations are required before the norm of the gradient is less than 10^{-10}?

. . . .

Exercise: How does the different Monte-Carlo simulation methods affect the statistical properties? What are the consequences for the optimal holdings?

. . . .

. . . .

Exercise: What are the optimal investments given Latin Hypercube sampling?

. . . .

Given the lognormal probability distribution, there exist a non-zero probability that the asset price can get arbitrarily close to zero or infinity.
Preparation: What are the consequences for the objective with $\gamma < 1$ for the case with shorting or borrowing?

. . . .

To avoid this issue in the unbounded solver, it is preferable to replace the estimation of the expected return with a more realistic value that gives an optimal solution that does not include shorting or borrowing. This can be achieved with CAPM since it implies that the market portfolio is optimal for the Mean-Variance model, and the power utility can be approximated with the Mean-Variance model.
Preparation: Given the covariance matrix C, the market capitalization weights w_M and the excess return $\mu_M - (e^r - 1)$ determine the expected return μ with CAPM. Remember from the proof of CAPM (TPPE33) that $\beta = \frac{Cw_M}{w_M^T C w_M}$. What is the relationship between the expected logarithmic return ν and the expected return μ?

. . . .

. . . .

Exercise: Compute the expected logarithmic return ν given that the market capitalization weights are the same for all assets.

Exercise: Given objective function values from optimal SP solutions and feasible solutions, compute the 95% confidence intervals for the upper and lower bounds.

Exercise: Which scenario generation method performs best? Which number of evaluations and number of scenarios gives good confidence intervals?

. . . .

. . . .

Table 1: Functions that are used to generate scenarios.

loadExcelFile	Read historical share prices from Excel
determineActiveReturns	Determine relevant historical values
estExpected	Calculates the yearly historical return
estVolEWMA	Calculate the yearly volatility and correlation with EWMA
genScenariosRegular	Generate scenario returns
genScenariosAntithetic	Generate scenario returns with antithetic sampling
genScenariosLatin	Generate scenario returns with Latin hypercube sampling
estStatistics	Determine the quality of scenarios
runUbounded	Main file