ORF435 / ORF535 / FIN535

Homework 3

Instructor: Professor John M. Mulvey

Due date: Thursday, 10/12/2017

Question 1: Utility Function and Markowitz Portfolio Optimization

(A) Show that the following properties on $U(x)$ are equivalent:

- U is risk-averse (chapter 11.3);
- $a(x) > 0$ where a is the Arrow-Pratt absolute risk-aversion coefficient;
- $C \leq E(d)$ where C is the certainty equivalent of a random payoff d.

(B) Compare $a(x)$ for log-utility function and quadratic utility function.

(C) To illustrate (B) with an example, consider a risky investment that has 60% probability of earning 5 (units) and 40% probability of losing 5. What is your investment decision for log-utility ($U(x) = \ln(x)$) and quadratic utility ($U(x) = x - \frac{1}{50}x^2$), if your initial wealth is 10? Or if your initial wealth is 20?

(D) Compute the certainty equivalent under each case in (C).
What do you think makes this difference? Comment on the risk aversion (usually called downside risk control in risk management) of the utility functions.

Computing Software Preparation

We will build our toolkit for asset allocation and risk management starting from this homework. The coding assignments will be very general scientific computing and optimization, so any language capable of these tasks will be sufficient.

Here we emphasize three mainstream languages in the academic research field: Python, MATLAB, and R. Optimization in Python can be fulfilled in the **scipy** package (`https://docs.scipy.org/doc/scipy/reference/tutorial/optimize.html`). For MATLAB, you can use the "fmincon" function. The documentation of fmincon (`https://www.mathworks.com/help/optim/ug/fmincon.html`) provides solid examples. Optimization in R is a little complicated, `https://cran.r-project.org/web/packages/optimization/optimization.pdf` might be a good start.

Much of our optimization problems, however, will be convex. In that sense, applying a convex solver will yield faster and more accurate results. **CVX** is a solver that can be implemented in MATLAB and Python; Gurobi is another one that can be implemented in more platforms. CVX for MATLAB can be downloaded at `http://cvxr.com/cvx/download/`, and the instructions can be found at `http://cvxr.com/cvx/doc/install.html`. CVXPY (`http://www.cvxpy.org/en/latest/`) is the Python version of cvx. Gurobi can be accessed via `http://www.gurobi.com` with the application of a student account.

Question 2: Classical Markowitz Model

1. Read the monthly return of three assets (stock, bond and T-bill) into your software.

2. Compute the arithmetic mean, geometric mean, annualized geometric mean, volatility (=standard deviation) and correlation matrix of the assets. Report the values.

3. Formulate the Markowitz portfolio optimization problem:
Minimize the portfolio variance given the budget constraint, non-negativity constraint (all weights must be positive, i.e. no shorting), return constraint (the **annualized** expected return of the portfolio should be greater than or equal to a given number $\bar{r}$).

4. Solve the (convex) problem above for $\bar{r} = [5\%, 6\%, 7\%, 8\%, 9\%, 10\%, 11\%, 12\%]$. Plot the annualized expected return and annualized volatility of these portfolios on a graph. You will get something like figure 6.11(b) in the book. This is called an **efficient frontier**.

5. Finally, find the portfolio that maximizes the Sharpe ratio. Notice that this is not a convex problem. Report the annualized expected return and annualized volatility of this portfolio and mark it on the efficient frontier.

Note: This part can be submitted with the next homework.