Portfolio Selection Optimization Models and Solution Approach

Abstract

Portfolio optimization is a very important area for long-term investors. It is concerned with the problem of how to best diversify investment into different classes of assets (such as stock, bonds, real estate, and options) in order to meet liabilities and to maximize the expected profit, while minimize the unacceptable risk. Portfolio optimization problems are based on mean-variance models for returns and for risk neutral density estimation. I focus on the asset and liability management (ALM) model which involves the stochastic programming, dynamic property, and nonlinear programming model based on scenario tree. The most challenging work to solve this problem is that the for a realistic model description the size of scenario tree quickly reaches astronomical sizes. To solve this problem efficiently, there are two major methods: decomposition method and interior point method. I will analyze how to apply the Interior Point method (IPM) to solve the optimal solution. Finally, I use the Matlab to simulate the algorithm and solve a relatively small size problem modeled by some history data from the website.

1. Classic Portfolio model

There are many different formulations of the portfolio problem have been modeled over time, starting from the Harry Markowitz’s seminal mean-variance model [1]. We have studied the model in the class. This model is seems to be of high importance. In the Markowitz model the investor has a choice between different investments or assets \( j \in A \). The return of each investment is given by a random variable \( R_j \), and the expectation of the return is \( \mu_j = E[R_j] \) of each investment and their joint covariance structure \( Q = \text{Cov}[R], Q_{ij} = \text{Cov}(R_i, R_j) \) are known. In the Markowitz model the twin contradictory goals of maximizing the expected portfolio return over all feasible portfolios \( E[R(x)] \), while minimizing the risk \( \text{Var}[R(x)] \). The problem can be denoted by that:

\[
\begin{align*}
\text{Minimize} & \quad x'Qx \\
\text{Subject to} & \quad E[R(X)] = m \quad (m \text{ is expected return value})
\end{align*}
\]

Alternatively, we can also formulate the problem with combing the expected portfolio return and risk into a single combined objective

\[
\begin{align*}
\text{Max} & \quad E[R(x)] - \lambda \text{Var}[R(x)].
\end{align*}
\]

While the Markowitz model has become an industry standard, it suffers from important shortcomings, relating lack of a dynamic structure. Being a one period model, there are many perceived weakness that emerged with over years: the assumption of normal asset returns, the assumption of known fixed means and covariance, and not least the inability to capture dynamic effects such as transaction costs and the possibility to hedge risk through rebalancing of the portfolio at the future time stages[2].

2. ALM (Asset and Liability Management) Model
The major applications for dynamic portfolio optimization are Asset and Liability Management (ALM) models in which the investor seeks an optimal long-term investment policy that meets anticipated (but unknown) liabilities and maximizes the expected surplus return, while minimizing the risk of defaulting on the liability payments. This is a model of prime importance to long-term investors such as insurances and pension funds.

Stochastic programming provides an appropriate framework for mathematical description of ALM models. Stochastic programming deals with the situation in which some of the data describing an optimization model is uncertain. This methodology can be described as an alternating sequence of decision and random realizations that occur at different points in time stages. For discrete case the stochastic process can be represented as a scenario tree[3].

![Scenario tree](image)

**Fig1 Scenario tree**

Each level \( t \) of the tree corresponds to a point in time when a realization of the random process becomes known and a subsequent decision is taken. Each node \( i \) corresponds to a particular series of events to this point in time. The branches from a particular node represent the (discrete set) of possible future outcomes of the random variables \( \xi_{t+1} \) in the next period. The root node of the tree represents the current time \( t = 0 \), and the leaf nodes represent the possible states of the system at the end of the planning horizon \( t = T \). We denote by \( V_t \) the set of nodes at level \( t \) in the tree and \( V = \bigcup V_t \) the complete node set. For every node \( i \in V_t \), we will denote by \( \pi(i) \in V_{t-1} \) its immediate ancestor and by \( C(i) \subset V_{t+1} \) its set of child nodes. For every node there is a transition probability \( \omega_{i} \) of reaching this node given that its parent \( \pi(i) \) has been reached. The total probability \( p_i \) of reaching node \( i \) is obtained by the product of all transition probabilities on the path from the root to node \( i \).

Every scenario, that is a path through the tree from the root to a leaf, represents a particular sequence of realizations \( (\hat{\xi}_1, \ldots, \hat{\xi}_T) \) of the random process \( \xi \), that is one particular outcome of the random data in the problem. Every node further carries its own version \( x_i \) of the decision variable \( x_t \) of the appropriate stage. For a realistic model description the size of the scenario tree quickly reaches astronomical sizes. The number of nodes is exponential in the number of time stages considered.

An ALM model is concerned with finding the optimal way of investing into \( J \)
assets \( j \in A \) over multiple time periods \( t = 0, \ldots, T \). The returns \( r^t \) of the assets in each time period are assumed to be uncertain, but based on some (known) random distribution. An initial amount of cash \( b_0 \) is invested at \( t = 0 \) and the portfolio may be rebalanced at discrete times \( t = 1, \ldots, T \), incurring transaction costs. At every time \( t \) a liability payment \( L_t \) of uncertain amount is due. The objective is to maximize the expectation of the final value of the portfolio at time \( T \) and to minimize the associated risk measured, for example, with the variance of the final wealth. In the stochastic programming formulation the evolution of the uncertain process driving the asset returns \( r^t = (r^t_j) j \in A \) is described by a scenario tree. Let \( v_j \) be the value of asset \( j \), and \( ct \) the transaction cost (expressed as a percentage of transaction volume). It is assumed that the value of the assets will not change throughout time and a unit of asset \( j \) can always be bought for \((1+ct)v_j\) or sold for \((1-ct)v_j\). Instead a unit of asset \( j \) held in node \( i \) (coming from node \( \pi(i) \)) will generate extra return \( r_i, j \).

**Model Variables:**
We denote by \( x^b_{i,j} \) the units of asset \( j \) held at node \( i \) and by \( x^b_{i,j}, x^s_{i,j} \) the transaction volume (buying, selling) of this asset at this node. We assume that we start with zero holding of all assets but with funds \( b_0 \) to invest and the value of each asset doesn’t change along the time. Further we assume that one of the assets represents cash, i.e., the available funds is always fully invested.

**Model Constraints:**
The standard constraints on the investment policy can be expressed as follows: cash balance constraints describe possible buying and selling actions within a scenario while taking transaction costs into account. The net cash flow in each node originating from selling and buying assets must be equal to the liability payments \( L_i \) in this node

\[
\sum_{j \in A} (1+c_i)v_j x^b_{i,j} - \sum_{j \in A} (1+c_i)v_j x^s_{i,j} = L_i \forall i \in V \setminus \{0\}
\]

\[
\sum_{j \in A} (1+c_i)v_j x^b_{0,j} = b_0.
\]

Each scenario is linked to its parent through inventory constraints; these are balance constraints on asset holdings (taking into account the random return on asset):

\[
(1 + r_{i,j}) x^b_{\pi(i),j} = x^b_{i,j} - x^b_{i,j} + x^s_{i,j} \quad \forall i \in V \setminus \{0\}
\]

**Model Objective**
We consider an objective function that maximizes the expected portfolio surplus return while take risk into account over all scenarios. The wealth of the portfolio in node \( i \in V_T \) at final time \( T \) is given by

\[
W_{r,i} = \sum_{j \in A} (1-c_i)v_j x^b_{i,j}
\]

The expected value of the portfolio at \( t = T \) is thus
\[ W_f = E[w_T] = \sum_{j \in V_T} p_j w_{T,j} = (1-c_t) \sum_{j \in V_T} p_j \sum_{j \in A} v_j x_{i,j}^h \]

If we take the portfolio risk into consideration, for computation simplicity, we may introduce an explicit variable \( y = \mathbb{E}[w_T] \) together with the constraint

\[ y = (1-c_t) \sum_{j \in V_T} p_j \sum_{j \in A} v_j x_{i,j}^h \]

resulting the model of formulation:

\[
\begin{align*}
\max_{x, y \geq 0} & \quad y - \lambda \left( \sum_{j \in V_T} p_j (1-c_t) \left( \sum_{j \in A} v_j x_{i,j}^h \right)^2 - y^2 \right) \\
\text{s.t.} & \quad (1 + \eta_{ij}) x_{i,j}^h = x_{i,j}^h - x_{i,j} + x_{i,j}^t, \quad \forall i \in V \setminus \{0\} \\
& \quad \sum_{j \in A} (1 + c_t) v_j x_{i,j}^h - \sum_{j \in A} (1 + c_t) v_j x_{i,j}^t = L_i, \quad \forall i \in V \setminus \{0\} \\
& \quad \sum_{j \in A} (1 + c_t) v_j x_{i,j}^h = b_i.
\end{align*}
\]

Defining the lattice matrices \( Q_i \),

\[
Q_i \in \mathbb{R}^{3 \times 3} : \begin{cases} 
Q_{ii,3k} = p_i (1-c_t)^2 v_j v_k, & f, k \in A, i \in V_T \\
Q_i = 0, & i \notin V_T
\end{cases}
\]

that have entries only in elements with a row and column index divisible by 3 (corresponding to the \( x_i^h \) variables). To illustrate the structure of this problem we gather decision vector components \( x_i \) for each node as

\[ x_i = (x_{i,1}^s, x_{i,1}^b, x_{i,1}^h, \ldots, x_{i,j}^h, x_{i,j}^h, x_{i,j}^h) \]

and define matrices

\[
W = \begin{pmatrix}
1 & -1 & 1 \\
& \ddots & \ddots & \ddots \\
& 1 & -1 & 1 \\
-c_i^s & c_i^b & 0 & \cdots & -c_j^s & c_j^b & 0
\end{pmatrix}, \quad T_i = \begin{pmatrix}
0 & 0 & 1 + r_{i,1} \\
& \ddots & \ddots & \ddots \\
0 & 0 & 0 & \cdots & 0 & 1 + r_{i,j} \\
0 & 0 & \cdots & 0 & 0 & 0
\end{pmatrix}
\]

and

\[ d_i \in \mathbb{R}^{3 \times 3} : (d_i)_{3j} = (1-c_t) p_j v_j, \]

Where \( c_j^b = (1 + c_t) v_j, c_j^s = (1 - c_t) v_j \). We can rewrite the problem as a compact form as below:
\[
\max_{x,y} \quad y - \lambda \left[ \sum_{i \in V} x_i^T Q, x_i - y^2 \right]
\]

subject to
\[
\sum_{j \in V} d_j^T x_j = y
\]

\[
T_i x_{e(i,j)} + W x_j = L_i e_{(j+1), e(i,j)}, \forall i \in V \setminus \{0\}
\]

\[
W_0 x_0 = b_0 e_{(j+1)}
\]

Where the \( e_{(J+1)} \) is the \( (J+1) \)-th unit vector \( e_{(J+1)} = (0, ..., 0, 1) \)'s. While the representation is very compact, it should be keep in mind that ALM problem can grow to enormous size. For example given \( T=5 \) time periods, 60 different assets and 30 branches at every node, resulting problem would have 24 million scenarios and \( 4.5 \times 10^9 \) variables. How to solve the problem efficiently is a hard problem.

3. Solution Approach: Interior Point Method Algorithm (IPM)

One of the most successful methods for the parallel solution of stochastic programming problems is IPMs. There are various reasons for this popularity: their applicability to a wide range of formulations spanning linear, quadratic, and nonlinear models, their comparative nonsensitivity to large problem sizes (IPMs are in practice observed to converge in \( O(\log N) \) iterations, where \( N \) is the problem size), and not least the amenability of the linear algebra operations to parallelization. For these reasons we will try to use interior point method to solve the portfolio optimization problem formulated before[4].

There are various different variants of IPMs, the most popular being primal–dual and primal IPM. They differ in details of the algorithm logic. We will restrict our attention to the primal–dual IPM. Further IPMs can be applied with minor modifications to linear, quadratic, and nonlinear optimization problems. Consider the quadratic programming problem:

\[
\min c^T x + \frac{1}{2} x^T Q x
\]

subject to
\[
Ax = b
\]

\[
x \geq 0,
\]

If we have some initial point \( x_0 \geq 0, s_0 \geq 0, p_0 \), which is not necessarily feasible for either primal or the dual problem, ie., \( Ax_0 \neq b \) and or \( A^T p_0 + s_0 - Q x_0 \neq c \). The Newton direction is obtained by solving the system of linear equations:

\[
\begin{bmatrix}
A & 0 & 0 \\
-Q & A^T & I \\
S & 0 & X
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta p \\
\Delta s
\end{bmatrix}
= \begin{bmatrix}
b - Ax^k \\
c - A^T p - s + Qx^k \\
ue - XSe
\end{bmatrix}
\]

We can converge to the optimal solution by using the Newton direction to iterate. Thus we can get the optimal solution to the original problem.

For an general nonlinear problem

\[
min f(x) \text{ s.t. } g(x) = 0, \ x \geq 0
\]

We need to use \( A = \nabla g(x), \ Q = \nabla^2 f(x) + \sum_{i=1}^m p_i \nabla^2 g(x) \) and the systems are
equivalent.

4. Application of ALM Model and Solution Approach

I apply the ALM model to the problem of the optimization portfolio of blue chips (Coca Cola), Hi-Tech corporation’s shares (Microsoft), real estate (PIMCO Real Estate) and Treasure bonds (Dexia bonds).

The scenario tree of this problem is shown as below:

Scenario tree
(The required data can be found from the Bloomberg’s website)

The transaction cost of each asset $c_t=0.2\%$.

The initial available cash is $b_0=100000$.

The liability $L_i$ at each node is a random variable with uniform distribution $[4000, 5000]$.

Asset price

<table>
<thead>
<tr>
<th>Asset</th>
<th>Asset value</th>
</tr>
</thead>
<tbody>
<tr>
<td>blue chips (Coca Cola)</td>
<td>$18.82/unit</td>
</tr>
<tr>
<td>Hi-Tech corporation’s shares (Microsoft)</td>
<td>$11.22/unit</td>
</tr>
<tr>
<td>real estate (PIMCO Real Estate)</td>
<td>$15.86/unit</td>
</tr>
<tr>
<td>Treasure bonds (Dexia bonds)</td>
<td>$42.897/unit</td>
</tr>
</tbody>
</table>

Return percentage of each node

<table>
<thead>
<tr>
<th>Asset(j)/node(i) ($r_{ij}$)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coca Cola</td>
<td>18.24</td>
<td>12.12</td>
<td>2.62</td>
<td>10.42</td>
<td>15.32</td>
<td>5.26</td>
</tr>
<tr>
<td>Microsoft</td>
<td>12.24</td>
<td>35.07</td>
<td>-10.62</td>
<td>-7.43</td>
<td>19.16</td>
<td>23.46</td>
</tr>
<tr>
<td>PIMCO Real Estate</td>
<td>8.23</td>
<td>8.96</td>
<td>8.05</td>
<td>7.29</td>
<td>8.35</td>
<td>8.34</td>
</tr>
<tr>
<td>Dexia bonds</td>
<td>8.12</td>
<td>8.26</td>
<td>9.11</td>
<td>8.95</td>
<td>8.34</td>
<td>9.01</td>
</tr>
</tbody>
</table>

Problem analysis:

By applying the ALM model to this problem, we can get that there are 85 variables in
this problem. The $A$ is a $36 \times 85$ matrix, the $Q$ is a $85 \times 85$ matrix. The system equation to solve the Newton direction is a $206 \times 206$ matrix. The structures of the matrices are shown as below. As we can see from those matrices, they are all sparse matrices.

\[
Q = \lambda \begin{bmatrix}
Q_6 \\
Q_5 \\
Q_4 \\
Q_3 \\
Q_2 \\
Q_1 \\
Q_0 \\
-1
\end{bmatrix}
\]

\[
Q_0 = 0 , \quad Q_1 = 0 , \quad Q_2 = 0
\]

\[
Q_3 = 0.6 \times 0.7 \times (1 - ct)^2
\]

\[
Q_4 = 0.3 \times 0.4 \times (1 - ct)^2
\]
\[ Q_5 = 0.4 \times 0.2 \times (1 - \alpha_t)^2 \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_1 \times v_1 & 0 & 0 & v_1 \times v_2 & 0 & 0 & v_1 \times v_3 & 0 & 0 & V_1 \times v_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_2 \times v_1 & 0 & 0 & v_2 \times v_2 & 0 & 0 & v_2 \times v_3 & 0 & 0 & v_2 \times v_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_3 \times v_1 & 0 & 0 & v_3 \times v_2 & 0 & 0 & v_3 \times v_3 & 0 & 0 & v_3 \times v_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_4 \times v_1 & 0 & 0 & v_4 \times v_2 & 0 & 0 & v_4 \times v_3 & 0 & 0 & v_4 \times v_4 \end{bmatrix} \]

\[ Q_6 = 0.8 \times 0.2 \times (1 - \alpha_t)^2 \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_1 \times v_1 & 0 & 0 & v_1 \times v_2 & 0 & 0 & v_1 \times v_3 & 0 & 0 & V_1 \times v_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_2 \times v_1 & 0 & 0 & v_2 \times v_2 & 0 & 0 & v_2 \times v_3 & 0 & 0 & v_2 \times v_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_3 \times v_1 & 0 & 0 & v_3 \times v_2 & 0 & 0 & v_3 \times v_3 & 0 & 0 & v_3 \times v_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_4 \times v_1 & 0 & 0 & v_4 \times v_2 & 0 & 0 & v_4 \times v_3 & 0 & 0 & v_4 \times v_4 \end{bmatrix} \]


\[ W = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & 1 \\ -18.8012 & 18.8388 & -11.2148 & 11.2372 & -15.8411 & 15.8759 & -43.9370 & 43.0230 \end{bmatrix} \]
After obtaining the all information about the model, I can use them to solve the optimal solution by using the interior point method. The critical process to get the optimal solution is to solve the Newton direction by solving the equation

\[
R = 
\begin{bmatrix}
1+r_{i,1} & 0 & 0 & 0 \\
0 & 1+r_{i,2} & 0 & 0 \\
0 & 0 & 1+r_{i,3} & 0 \\
0 & 0 & 0 & 1+r_{i,4}
\end{bmatrix}, i=1,2,3,4,5,6
\]

I use matlab to write a primal-dual interior method program to solve the problem automatically, and I get the following result: The X axis denotes the 85 decision variables, and the Y axis denotes the number of units corresponding to the decision variables.

For \( \lambda = 0.1 \), the optimal expected return is $10729.14 and the decision variables are shown as figure below:

For \( \lambda = 0.5 \), the optimal expected return is $10318.14 and the decision variables are shown as figure below:
For $\lambda = 1$ the optimal expected return is $10203.21$ and the decision variables are shown as figure below:

**Solution Analysis:**

By observing the solution above, we can see that as $\lambda$ increases, the expected return will decrease. They are reasonable solutions. Intuitively, the investor can't achieve both high return and low risk. So, as the coefficient of the covariance increase, it means that the investor consider more about the risk effect, resulting the expected return will decrease. What I have done is to seek an optimal strategy to balance the high return and different risk requirement.

**5. Conclusion**
Optimization models play an important role in financial decisions. The classical one stage model is **standard Markowitz portfolio model** which is the base of the other extension models. But it has some weakness that can’t capture the character of realistic model. By extend single period to multi-periods; I introduce the Asset and Liability Management (ALM) model. ALM model is formulated by the stochastic programming framework. While the ALM model provide a better tool to model a wide spectrum of realistic issues in portfolio allocation, the variables of the model grow to astronomical sizes very quickly in realistic model. An efficient method to overcome the problem is interior point method (IPM). Because it is applicable to a wide range of formulations spanning linear, quadratic, and nonlinear model; it is non-sensitive to large problem size; and so on. To test the ALM model and the IPM method, I find some portfolio data from the Bloomberg website and formulate a realistic ALM model. Then, I use IPM method to solve the problem and get a reasonable optimal solution.

6. References:
4. Dimitri P. Bertsekas, Nonlinear Programming