CFRM 507 Project 3 Part 1: Formulation

The unit of money is thousand dollars in the following illustration.

(1) Target retirement account balance

The investment account must provide an amount equal to 60% of the pre-retirement income:

$$103.63 - 0.03(55 - 66) * 1 * 60\% = 62.376$$

They plan to spend 3% of their account balance:

$$target\ account\ balance\ * 3\% \ = \ 62.376$$

Therefore, the target retirement account balance is 2079.2.

(2) Annual savings amounts each year until retirement for each client

Savings amounts = (103.63-0.03(55-current age))*0.15

Therefore, the annual savings amounts each year for each client are:

Amy Abrams: [15.5265, 15.531, 15.5355, 15.54, 15.5445, 15.549, 15.5535, 15.558, 15.5625, 15.567, 15.5715, 15.576, 15.5805, 15.585, 15.5895, 15.594]

Bob Brown & Carla Clausen: [15.5445, 15.549, 15.5535, 15.558, 15.5625, 15.567, 15.5715, 15.576, 15.5805, 15.585, 15.5895, 15.594]

Darrin Dorne: [15.558, 15.5625, 15.567, 15.5715, 15.576, 15.5805, 15.585, 15.5895, 15.594]

Eric Evans: [15.5805, 15.585, 15.5895, 15.594]

Francine Farnsworth: [15.594]

(3) The constant rate of return r

$$current\ account\ balance * (1 + r)^{67 - current\ age} + \sum_{i=1}^{67 - current\ age - 1} annual\ savings (1 + r)^{67 - current\ age\ - i} = target\ account\ balance$$

Take Francine Farnsworth as an example:

$$1600(1 + r)^2 + 15.594(1 + r) = 2079.2$$

MATLAB can solve this equation:

```
>> eq = '1600*(1+r)^2+15.594*(1+r)=2079.2';
>> s = solve(eq);
```

Client	Amy	Bob	Carla	Darrin	Eric	Francine
Return	0.033	0.055	0.099	0.025	0.108	0.135

(4) One-year delay

If retirement is delayed by one year, each client will save one more contribution to the savings account and the amount is

$$103.63 - 0.03(55 - 67) * 1 * 15\% = 15.5985$$

The method to calculate the constant rate is similar. Take Francine Farnsworth as an example again:

$$1600(1+r)^3 + 15.594(1+r)^2 + 15.5985(1+r) = 2079.2$$

Client	Amy	Bob	Carla	Darrin	Eric	Francine
Return	0.031	0.050	0.091	0.022	0.087	0.085

(5) Formulation

Parameters:

According to the data, we can calculate the return and the portfolio variance of each mix.

Assume that the weight of each asset j in mix i is w_{ij}. The weight vector of each mix i is w_i and the covariance matrix is Σ.

$$r_i = w_{ij} * r_j$$
$$var_i = w_i^T \Sigma w_i$$

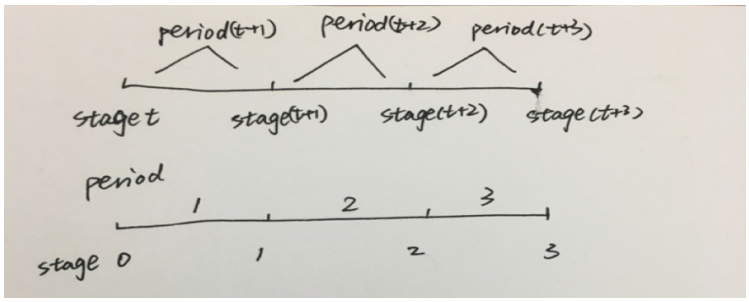
The above mean and standard deviation of each mix is log-normal. With the relationship of normal distribution and log-normal distribution, we can calculate the mean and std of $\ln(1+r_i)$. $\sum \ln(1+r_i)$ is normal distribution as well. With the mean and std of $\sum \ln(1+r_i)$, we can get the mean return and std of $\prod(1+r_i)$.

Every customer will have their own investment plan for each year so that they have their own optimization solution. Each time we solve the problem, we need to change the data of the age and account balance for each client. In other words, we need to solve the optimization problem for six times. Therefore, the following formulation takes Amy Abrams as an example to explain it better.

c: the current account balance. For example, the current account balance of Amy Abrams is 1000.

f_i: the fee rate of mix i. for example, $f_1 = 20bp$.

It is a combination of dynamic programming and stochastic programming with multiple periods and adaptive decisions. At each stage t, we need to decide which mix to invest and during each period, it will have a return. All the clients want to maximize their utility function.



Variables:

Stage t: t = current age of the client+1, current age of the client+2, …, 67

$m_{i,t}$: Mix i which will be invest in period t. $1 \le i \le 8$.

R($m_{i,t}$): the return in period t when invest Mix i. It is decided by $m_{i,t}$.

x_t: the account balance at stage t. $x_{current\ age} = current\ account\ balance$.

Objective function: maximize the expected utility

Maximize: $E(U(x))$ and U(x) is given in the project statement. With Monte Carlo simulation in the scenario-based approach, it can approximate the probability of each x to calculate the expectation of utility function.

Constraints:

$$|m_{i,t} - m_{i,t-1}| \leq 1$$

This constraint can avoid large jumps for investment mix.

Recursion constraint: $x_t = x_{t-1}(1 + \mathrm{R}(m_{i,t}))(1\text{-}f_i) + 0.15(103.63 - 0.33(55 - t))$

We can consider the problem in a formulation of **dynamic programming**.

Stage: age t (current age to 67)

State: account balance x_t and invest mix $m_{i,t-1}$ for last time

Decision: invest in mix $m_{i,t}$ and there is a constraint for decision:

$$|m_{i,t} - m_{i,t-1}| \leq 1$$

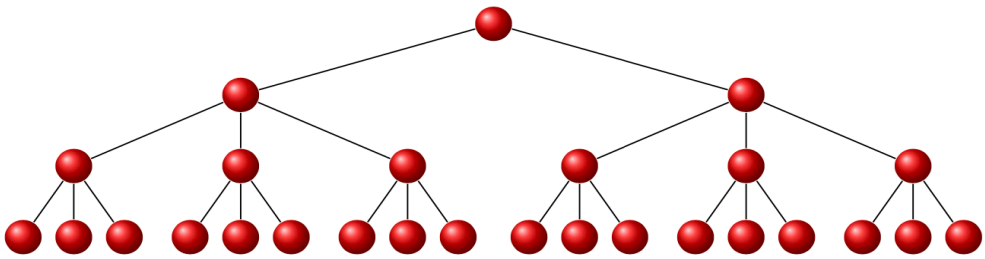
Recursion constraint:

$$E(U(t, x_t)) = \sum P(x_{t+1}) * U(t+1, x_{t+1})$$

$$x_t = x_{t-1}(1 + \mathrm{R}(m_{i,t}))(1\text{-}f_i) + 0.15(103.63 - 0.33(55 - t))$$

(6) Solution method description

The basic idea to solve this problem is a scenario-based approach. For example:



It is a typical data structure: tree. It is important to decide the number of the children for each parent's node. A large number will result in an unaffordable time while a small number will lead to inaccuracy. It needs adjustment in the programming process. For example, we can try 50, 100, 150 and so on to figure out some rules for the final result. However, in some sense, the growth rate is exponential so that when the stage is large enough, it will be troublesome. One way of solving the problem is to apply the neutral gas method to build to reduced scenario tree for multi-stage stochastic programming. The buildtree and checktree functions in the scenario R package may be helpful[1].

[1] https://cran.r-project.org/web/packages/scenario/vignettes/buildtree.html

Another solution is that we can use a smaller number for later generations. Scenario generation method includes conditional sampling, property matching methods, model-based cutting and growing method and so on. The return is log-normal distribution so that sampling method can be applied. It is easy to implement and distribution converges to the real one. Model-based cutting and growing method is useful for linear stochastic programming and it can reduce scenarios by cutting branches.