CFRM 507
Rohan Tiwari
Final Project Problem Formulation

Introduction

We are responsible for managing retirement accounts of several clients. Our goal is to provide each with a recommended portfolio allocation as well as a description of the likely consequences of following our advice. We need to decide how to invest their savings so that their account balance can grow to meet or exceed a target amount at the time of retirement which will be their 67th birthday.

The target retirement account balance

We are told that each client aspires to live on 90% of their preretirement income and Social Security and other savings will provide 30% of that amount. This implies the investment account must provide an amount equal to 60% of the pre-retirement income. Client's desired retirement income level is reached on their 66th birthday. Therefore, target account balance (tab) is given by

$$tab = \frac{(95000 - 0.02 * (50 - 66)^2 * 1000) * 0.6}{0.035} = 1540800$$

<u>The target retirement account balance is $1540.8K.</u>

The Annual Savings amounts each year from now until retirement for each client

Annual Savings amounts = 95000 – 0.02*(50 – CurrentAge)^2 * 1000.

Amy Abrams:

ages	Savings in thousands of dollars
50	14.25
51	14.247
52	14.238
53	14.223
54	14.202
55	14.175
56	14.142
57	14.103
58	14.058
59	14.007
60	13.95
61	13.887
62	13.818
63	13.743
64	13.662
65	13.575
66	13.482

Bob Brown:

ages	Savings in thousands of dollars
51	14.247
52	14.238
53	14.223
54	14.202
55	14.175
56	14.142
57	14.103
58	14.058
59	14.007
60	13.95
61	13.887
62	13.818
63	13.743
64	13.662
65	13.575
66	13.482

Carla Clausen:

ages	Savings in thousands of dollars
52	14.238
53	14.223
54	14.202
55	14.175
56	14.142
57	14.103
58	14.058
59	14.007
60	13.95
61	13.887
62	13.818
63	13.743
64	13.662
65	13.575
66	13.482

Darrin Dorne:

ages	Savings in thousands of dollars
53	14.223
54	14.202
55	14.175

56	14.142
57	14.103
58	14.058
59	14.007
60	13.95
61	13.887
62	13.818
63	13.743
64	13.662
65	13.575
66	13.482

Eric Evans:

ages	Savings in thousands of dollars
56	14.142
57	14.103
58	14.058
59	14.007
60	13.95
61	13.887
62	13.818
63	13.743
64	13.662
65	13.575
66	13.482

Francine Farnsworth:

ages	Savings in thousands of dollars
56	14.142
57	14.103
58	14.058
59	14.007
60	13.95
61	13.887
62	13.818
63	13.743
64	13.662
65	13.575
66	13.482

Giovanni Granville:

ages	Savings in thousands of dollars
56	14.142
57	14.103
58	14.058
59	14.007
60	13.95
61	13.887
62	13.818
63	13.743
64	13.662
65	13.575
66	13.482

Heloise Hart:

ages	Savings in thousands of dollars
58	14.058
59	14.007
60	13.95
61	13.887
62	13.818
63	13.743
64	13.662
65	13.575
66	13.482

Isaac Iverson:

ages	Savings in thousands of dollars
61	13.887
62	13.818
63	13.743
64	13.662
65	13.575
66	13.482

Jennifer Jones:

ages	Savings in thousands of dollars
64	13.662
65	13.575
66	13.482

Constant rate of return that individual would need to generate to accumulate the targeted amount

Let the constant rate of return be denoted by r.

(

For each client k, we can write the expression connecting target account balance (tab_k), current account balance(cab_k), annual savings($annualSavings_k$) and tomorrow's age (age_k) as follows:

$$tab_k = cab_k(1+r)^{67-age_k} + \sum_{i=age_k}^{67-age_k-1} annualSavings_i * (1+r)^{67-i}$$

We need to solve the above equation for each client to get the constant rate of return.

Client	Constant rate of return (r)
Amy Abrams	0.033079
Bob Brown	0.0447505
Carla Clausen	0.0488352
Darrin Dorne	0.109703
Eric Evans	0.0283216
Francine Farnsworth	0.0477588
Giovanni Granville	0.088927
Heloise Hart	-0.00604509
Isaac Iverson	0.062482
Jennifer Jones	-0.0208998

Jennifer Jones already has an account balance that is higher than the retirement target which explains the negative rate of return. Jennifer can choose to not invest their savings in any of the mixes depending on risk aversion and utility.

Heloise Hart's savings until retirement add up to more than the retirement target which explains the negative rate of return. Heloise can also choose to not invest their savings in any of the mixes depending on risk aversion and utility.

Rate of return necessary if retirement is delayed by one year

If retirement is delayed by one year, each client gets an additional year of savings, and the retirement target account balance will be updated. The new target account balance (tab) is given by

$$tab = \frac{(95000 - 0.02 * (50-67)^2 * 1000) * 0.6}{0.035} = 1529486$$

Therefore, the new target account balance is $1529486

The calculation of the constant rate of return will remain the same as before. For each client k, we can write the expression connecting target account balance (tab_k), current or initial account balance (cab_k), annual savings ($annualSavings_k$) and tomorrow's age (age_k) as follows:

$$tab_k = cab_k(1+r)^{68-age_k} + \sum_{i=age_k}^{68-age_k-1} annualSavings_i * (1+r)^{68-i}$$

We need to solve the above equation for each client to get the constant rate of return.

Client	Constant rate of return
Amy Abrams	0.0300139
Bob Brown	0.0407139
Carla Clausen	0.0442766
Darrin Dorne	0.0997896
Eric Evans	0.0243484
Francine Farnsworth	0.0419651
Giovanni Granville	0.079115
Heloise Hart	−0.00707409
Isaac Iverson	0.0505626
Jennifer Jones	−0.0196337

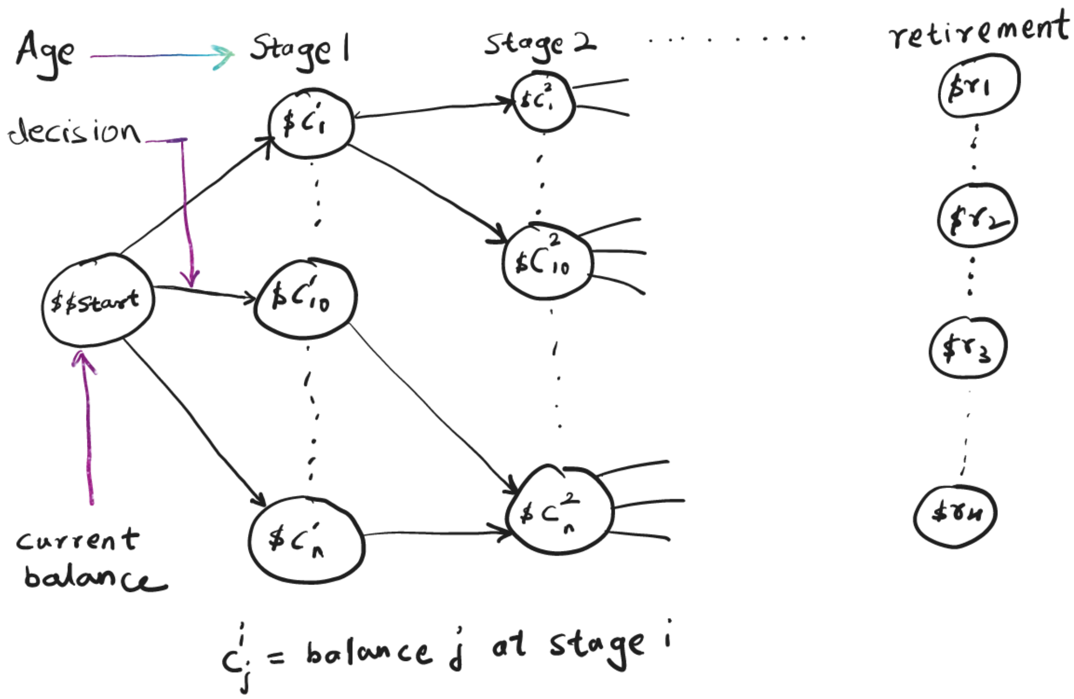
Methodology

Stage: Current age of the client in years.

State: The current account balance in thousands of dollars. This is determined by adding up the previous stage's account balance and the return on the mix invested in.

Decision: The mix to invest in at a given stage, state combination.

In this problem, we will consider Monte Carlo based market outcomes of the returns of different mix each year. We will build a scenario tree in which each node will denote the account balance, and each level of tree will denote a particular stage or age in years. Each level or stage will consist of many nodes that will denote the possible account balances that a client can achieve at that age depending on the decision taken at the previous age. The scenario tree spans from the youngest client's current age till the set retirement age.



The mix returns are jointly log-normally distributed. We can generate several return samples for the next 1-year for each mix and evaluate the resulting account balances if a particular mix for investment is chosen and the associated utility. To select the best mix at a given (age, account balance) combination, we will compare the expected utility across all the mixes and pick the best one with max utility.

The solution will be implemented using a combination of a scenario-based approach using randomized mix returns and dynamic programing (with backward recursion). Dynamic programming will help us by breaking the evaluation of the scenario tree into sub problems. Each sub problem's solution will give us the best mix to choose for a given age t and account balance c combination based on maximizing the expected utility. We can store the sub problem solutions in a data structure to facilitate quick look ups. More details are explained in the Constraints section.

Parameters

Retirement age in years: 67

Smallest age in the problem in years: 50. This is the age of the youngest client in our problem.

Target account balance: Tab = $1540.8K

Annual Savings rate: 0.15 or 15%

Number of assets: 7

Number of mixes: 7

Annual fee for mix i in basis points: F_i

Initial age of each Client k in years: age_k

Initial account balance of each client k in thousands of dollars: cab_k

Mean returns for each asset class j: μ_j

Standard deviation for each asset class j:σ_j

Covariance matrix for asset classes: Σ_a calculated as follows

Suppose correlation matrix for asset classes is denoted by ρ, then the covariance matrix is given by

$$\Sigma_a = diag(stddevs\_asset) * \rho * diag(stdevs\_asset)$$

Where $stddevs\_asset$ is the vector of stddevs for the assets. Diag(..) creates a diagonal matrix.

Weight distribution vector for each mix i: ω_i

Mean return for each mix i: μ_i

Standard deviation for each mix i:σ_i

covariance matrix for the mixes: Σ_M

Suppose correlation matrix for asset classes is denoted by q, then the covariance matrix is given by

$$\Sigma_M = diag(stddevs\_mixes) * q * diag(stdevs\_mixes)$$

Where $stddevs\_mixes$ is the vector of stddevs for all the mixes. Diag(..) creates a diagonal matrix

Salary function: As given in the problem statement.

Utility function: As given in the problem statement.

Number of normal return samples to generate per mix: s

Random annual returns vector of size s for each mix i: $Simulated\_returns_i$

Stage: Current age of the client in years denoted by t as we progress through the scenario tree year by year. The starting value is determined by the youngest client's age and the final value is the retirement age. Since these are known at the problem setup, we consider them as parameters.

State: The current account balance in thousands of dollars denoted by c. This is determined by adding up the previous stage's account balance and the return on the mix invested in the previous stage. We can calculate a reasonably large set of possible balances starting from a small balance until the retirement target balance and make this a problem parameter.

Variables

Decision: The best mix i to invest in each stage t and state c combination. This determines the edges in the scenario tree and is unknown when we start.

Payoff: The maximum expected utility for selecting the best mix i for a given stage t and state c combination which determines the optimal decision at each stage, state combination.

Constraints

We consider a set of balances starting from every age t. Let c_j denote a starting balance at age t in the scenario tree.

Let balance_deltas$_k$ denote a vector of all possible balances at age $t+1$, starting from age t with balance c_j after investing in a mix k. We can write this vector as:

balance_deltas$_{k,\,t+1}$ = $(c_j + savings(\,t\,)) * (1 + simulated_{returns_k}) * (1 - F_k)$

$simulated\_returns_k$: we start simulated normal returns (y_k) for mix k for the next 1 year and compute $e^{y_k} - 1$ and store in this vector.

F_k: the annual fee in basis points for mix k. $(1 - F_k)$ is the resulting return after subtracting the fee.

For each of these balances in balance_deltas$_{k,\,t+1}$, we need to calculate the utility and then the expected utility. These utility values can be found in the dynamic programming solution of the previous subproblem which is the solution for age $t+1$ and the particular balance in balance_deltas$_k$. Once we have these utilities, we can take the average to get the expectation.

So, the payoff for age t, balance c_j and decision (or mix k) is given as $\underset{s}{mean}[V(t,\ balance\_deltas_k,$ $d_k)]$ for a given mix.

This average is taken over all the possible balances in balance_deltas$_{k,\,t+1}$. The length of this vector is s which denotes the number of normal return samples simulated for mix k. d_k denotes the decision of investing in mix k.

We need to do this for all the 7 mixes and then calculate the max across all the 7 means to get the best resulting payoff and recursion constraint at age t, balance c_j and decision of best mix k to choose.

$$v\big(t, c_j, d_k\big) = \underset{k}{max}[\underset{s}{mean}\big[V\big(t+1, \quad balance\_deltas_{k,t+1}, \quad d_k\big)\big]$$

Objective function

At each stage and state, the decisions are taken such that the expected utility of the client is maximized at retirement. For each client, our goal is to maximize $E[u(x)]$ where $u(x)$ is given in the problem statement. Therefore, the objective function for the dynamic programming problem is given by:

For each client, maximize $E[u(x)]$

where $u(x)$ is given in the problem statement