Constrained Dynamic Quadratic Optimization

Fei Cong [a]   Kees Oosterlee [a,b]

[a]TU Delft

[b]Centrum Wiskunde & Informatica, Amsterdam

23 May 2016

Overview

1. Problem Formulation
2. Forward-backward Algorithm
3. Numerical Experiments

Dynamic Quadratic Optimization

- Value function:

$$J_0(W_0) = \max_{\{x_t\}_{t=0}^{T-\Delta t}} \left[\mathbb{E}[(W_T - \gamma)^2 | W_0] \right], \tag{1}$$

where $\{x_t\}_{t=0}^{T}$ are the asset allocations at sequential time steps and the wealth evolves following:

$$W_{t+\Delta t} = W_t \cdot (x_t \cdot R_t^e + R_f), t = 0, \dots, T - \Delta t.$$

- R_t^e: excess return of the risky asset in the period $[t, t + \Delta t)$
- R_f: return of the risk-free asset
- γ: pre-determined target

Solving Dynamic Quadratic Problem

- Using the Bellman principle, we can write the value function in a recursive version:

$$J_t(W_t) = \min_{x_t \in A} \left[\mathbb{E}[(J_{t+\Delta t}(W_{t+\Delta t}) | W_t] \right],$$

with $J_T(W_T) = (W_T - \gamma)^2$.

- However, it is still difficult to solve in the constrained case.

Solving Dynamic Quadratic Problem

- Using the Bellman principle, we can write the value function in a recursive version:

$$J_t(W_t) = \min_{x_t \in A} \left[\mathbb{E}[(J_{t+\Delta t}(W_{t+\Delta t}) | W_t] \right],$$

with $J_T(W_T) = (W_T - \gamma)^2$.

- However, it is still difficult to solve in the constrained case.

Unconstrained Optimization Problem

$$\min_{x_t} \left[\mathbb{E}[J_{t+\Delta t}(W_{t+\Delta t}) | W_t] \right]$$

- Smooth value functions $J_{t+\Delta t}(W_{t+\Delta t})$
- Optimal x_t can be obtained by solving

$$\frac{\partial \mathbb{E}[J_{t+\Delta t}(W_{t+\Delta t}) | W_t]}{\partial x_t} = 0$$

$$\Rightarrow \mathbb{E}\left[\frac{\partial J_{t+\Delta t}(W_{t+\Delta t})}{\partial x_t} | W_t \right] = 0.$$

Constrained Optimization Problem

$$\min_{x_t} \left[\mathbb{E}[(J_{t+\Delta t}(W_{t+\Delta t}) | W_t] \right]$$

- Non-smooth value functions

$$\frac{\partial \mathbb{E}[J_{t+\Delta t}(W_{t+\Delta t}) | W_t]}{\partial x_t} \neq \mathbb{E}\left[\frac{\partial J_{t+\Delta t}(W_{t+\Delta t})}{\partial x_t} | W_t \right].$$

- Optimize based on a derivative-free approach

Numerical Algorithm

- A forward solution: sub-optimal but very efficient
- Several backward updatings

A Forward Solution

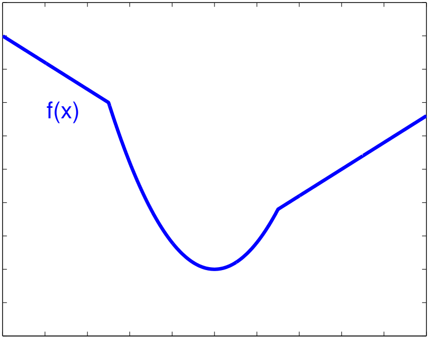
We only require the forward solution to be efficient. They can be:

- constant asset allocation
- myopic solution (assuming that the investor only invests optimally in the coming period and afterwards she will choose the risk-free strategy.)
- ...

Backward Recursive Programming

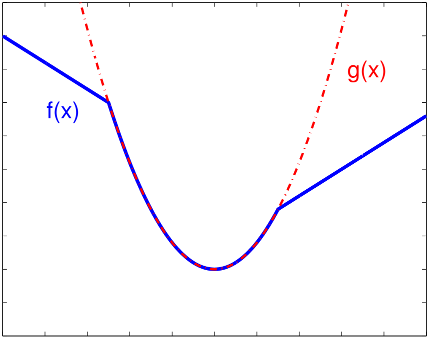
- We have a forward solution. In general, it is easy to generate but not optimal.
- To improve the solution, we consider the backward recursive programming.

A toy problem



$$\arg\min_x f(x) = ?$$

A toy problem



$$\arg\min_x f(x) = \arg\min_x g(x)$$

Benefit from suboptimal solution

$$J_t(W_t) = \min_{x_t \in A} \left[\mathbb{E}[(J_{t+\Delta t}(W_{t+\Delta t}) | W_t] \right]$$

Suppose that we know the optimal allocation should lie in a small area A_η instead of A, then the wealth $W_{t+\Delta t}$ under the optimal control should be located in the domain:

$$D_{t+\Delta t} := \{W_{t+\Delta t} | W_{t+\Delta t} = W_t \cdot (x_t \cdot R_t^e + R_f), \quad x_t \in A_\eta\}.$$

It implies that what we really care about is the following problem:

$$J_t(W_t) = \min_{x_t \in A} \left[\mathbb{E}[(J_{t+\Delta t}(W_{t+\Delta t}) | W_t, W_{t+\Delta t} \in D_{t+\Delta t}] \right],$$

Local Optimization

$$J_t(W_t) = \min_{x_t \in A} \left[\mathbb{E}[(J_{t+\Delta t}(W_{t+\Delta t}) | W_t] \right]$$

$$\Downarrow$$

$$J_t(W_t) = \min_{x_t \in A} \left[\mathbb{E}[(J_{t+\Delta t}(W_{t+\Delta t}) | W_t, W_{t+\Delta t} \in D_{t+\Delta t}] \right]$$

- The value function $J_{t+\Delta t}(W_{t+\Delta t})$ is non-smooth from the global perspective.
- Usually it is piece-wise smooth.
- For a local problem, we can parameterize the value function as a smooth function.

General Description of the Backward Programming

- We simulate paths using a forward strategy and locally improve the optimality in the backward recursion.
- In the process, we need to calculate conditional expectations. We use a regress-based method proposed in [Jain and Oosterlee, 2015].

Initial guess is not so important!

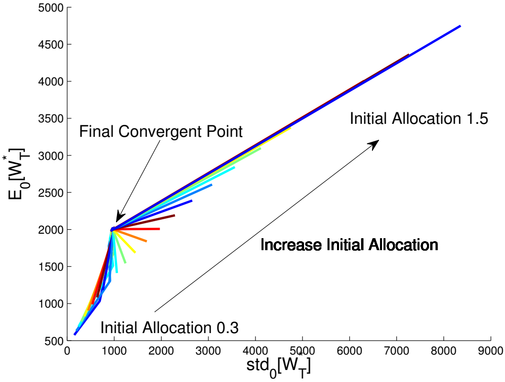


Figure: We start with fixed asset allocations and do iterative updating.

Summary and Future Work

- Summary:
  - Solve the constrained optimization problem with Monte-Carlo simulation
  - A forward sub-optimal solution
  - backward updating
- Future Work: Robust optimization

Jain, S. and Oosterlee, C. W. (2015).
The stochastic grid bundling method: Efficient pricing of Bermudan options and their Greeks.
Applied Mathematics and Computation, 269(1):412–431.

Thank you!