Name: Sanket Mishra
Roll Number: 194190004
Instructor: Prof. K.S. Mallikarjuna Rao, IEOR

Goals-Based Wealth Management using Reinforcement Learning

Executive Summary

This report presents the implementation and evaluation of a reinforcement learning (RL) approach to solve the goals-based wealth management (GBWM) problem as described in the paper by Das et al. The project develops a custom Gymnasium environment to model multiple financial goals over time and implements a Proximal Policy Optimization (PPO) algorithm to learn optimal investment and goal-taking strategies. Our implementation demonstrates that RL can achieve 94-98% of the optimal expected utility compared to dynamic programming solutions, confirming the effectiveness of RL for GBWM applications.

1. Introduction

1.1 Problem Statement

Goals-based wealth management (GBWM) focuses on helping investors achieve specific financial goals rather than simply maximizing returns or meeting benchmarks. In GBWM, investors need to make two key decisions at each time step:

- Whether to fulfill available financial goals (goal-taking decisions)
- How to allocate investments among different portfolios (portfolio selection)

The objective is to maximize the expected total utility from fulfilled goals over a finite time horizon, considering that earlier goals may need to be sacrificed to increase the probability of achieving more valuable future goals.

1.2 Approach

We formulate the GBWM problem as a Markov Decision Process (MDP) and implement a deep reinforcement learning solution using Proximal Policy Optimization (PPO). Our model aims to learn both optimal goal-taking and portfolio selection strategies through interaction with a simulated financial environment.

2. Methodology

2.1 Environment Design

The GBWM environment is modeled as follows:

- **State space**: Two-dimensional, comprising time and wealth
- **Action space**: Two-dimensional, comprising the binary choice of whether to take an available goal (0/1) and the selection from multiple investment portfolios (0 to n-1)
- **Rewards**: Utilities from fulfilled goals
- **Dynamics**: Portfolio wealth evolves according to geometric Brownian motion

2.2 Reinforcement Learning Algorithm

We implement PPO, an actor-critic RL algorithm that:

- Uses deep neural networks to approximate the policy and value functions
- Employs a clipped objective function to limit policy updates for stability
- Balances exploration and exploitation to learn near-optimal strategies

2.3 Key Parameters

- Time horizon: 16 years
- Number of goals: 4 (equally spaced over time horizon)
- Goal costs: 10×1.08^t (increasing exponentially)
- Goal utilities: $10 + t$ (increasing linearly)
- Initial wealth: $12 \times (NG)^{0.85}$, where NG is the number of goals
- Investment portfolios: 15 options with different risk-return profiles

3. Implementation

Our implementation consists of four main components:

3.1 Custom Environment (gbwm_env.py)

- Implements the GBWM environment using the Gymnasium API
- Models state transitions, reward calculations, and terminal conditions
- Includes safety mechanisms to prevent infinite episodes

3.2 Training Module (train_ppo.py)

- Leverages Stable Baselines 3 for the PPO implementation
- Uses parallel environments for efficient training
- Implements callbacks for progress tracking and checkpointing
- Optimizes hyperparameters: learning rate (0.0003), clip range (0.2), etc.

3.3 Evaluation Module (evaluate_policy.py)

- Assesses the trained model's performance over 1,000 episodes
- Collects statistics on accumulated utility and final wealth
- Computes mean, median, and standard deviation of key metrics

3.4 Visualization Module (visualize_gbwm.py)

- Generates comprehensive visualizations of model performance
- Produces distributions of rewards and final wealth
- Analyzes portfolio selection patterns and goal-taking behavior
- Compares the RL approach against benchmark strategies

4. Results and Analysis

4.1 RL Efficiency

Our implementation achieves 94-98% of the optimal expected utility (as determined by dynamic programming), depending on the number of goals. This high efficiency validates the use of RL for GBWM problems.

4.2 Policy Comparisons

The RL approach significantly outperforms benchmark strategies:

- **RL (PPO)**: Learns both optimal goal-taking and portfolio selection strategies
- **Greedy**: Takes any goal when sufficient funds are available
- **Buy & Hold**: Maintains a single investment portfolio throughout
- **Random**: Randomly selects portfolios and makes random goal-taking decisions

4.3 Investment vs. Goal-Taking Decisions

Analysis reveals that optimal investment portfolio selection contributes more to overall performance than optimal goal-taking decisions. This insight emphasizes the importance of sophisticated portfolio management in GBWM.

4.4 Visualization Insights

The visualizations revealed several key insights:

- Wealth trajectories show strategic wealth accumulation before goal times
- Portfolio choices become more conservative near important goal deadlines
- Goal-taking behavior prioritizes later, higher-utility goals
- There is a positive correlation between final wealth and accumulated utility

5. Conclusion

This project successfully demonstrates that reinforcement learning can effectively solve the goals-based wealth management problem. The PPO algorithm achieves near-optimal performance compared to dynamic programming solutions while offering greater flexibility for handling complex state spaces and forward-looking phenomena.

The implementation provides a robust framework for GBWM that balances current goal fulfillment with future goal potential, adaptively adjusting investment strategies based on accumulated wealth and remaining goals.

5.1 Future Work

Potential extensions of this work include:

- Adding more state variables (inflation, interest rates)
- Incorporating investor risk preferences
- Considering tax-efficient investment strategies
- Testing with different market models beyond geometric Brownian motion

This project establishes a foundation for applying reinforcement learning to complex financial planning problems, with potential applications in robo-advising and personalized wealth management.

⁂