G-Learner and GIRL:
Goal Based Wealth Management with Reinforcement Learning

abstract

- goal based wealth management problems such as optimization of retirement plans or target dated funds.
  - an investor seeks to achieve a financial goal
    - making periodic investments in the portfolio while being employed, and periodically draws from the account when in retirement
    - the ability to rebalance the portfolio by selling and buying different assets (e.g. stocks).
- G-Learner:
  - a reinforcement learning algorithm that operates with explicitly defined one-step rewards,
  - does not assume a data generation process,
  - suitable for noisy data.
  - based on G-learning (Fox et al., 2015)
    - a probabilistic extension of the Q-learning method of reinforcement learning
- demonstrate how G-learning
  - gives an entropy-regulated Linear Quadratic Regulator (LQR).
  - applied to a quadratic reward and Gaussian reference policy

abstract

- This critical insight provides
  - a novel and computationally tractable tool for wealth management tasks : to high dimensional portfolios
  - present a new algorithm, GIRL, that extends our goal-based G-learning approach to the setting of Inverse Reinforcement Learning (IRL) where rewards collected by the agent are not observed, and should instead be inferred
  - demonstrate that GIRL can successfully learn the reward parameters of a G-Learner agent and thus imitate its behavior.
  - Finally, we discuss potential applications of the G-Learner and GIRL algorithms for wealth management and robo-advising.

Introduction : MVO

- Mean-variance Markowitz optimization (MVO) (Markowitz, 1959)
  - remains one of the most commonly used tools in wealth management.
  - Portfolio objectives in this approach : expected returns and covariances of assets in the portfolio,
  - may not be the most natural formulation for retail investors.
- retail investors seek specific financial goals for their portfolios.
  - a contributor to a retirement plan may demand that the value of their portfolio at the age of his or her retirement be at least equal to, or preferably larger than, some target value P_T

Introduction : Goal based Wealth management

- Goal-based wealth management
  - offers some valuable perspectives into optimal structuring of wealth management plans
    - retirement plans or target date funds
  - The motivation for operating in terms of wealth goals can be more intuitive (while still tractable) than the classical formulation in terms of expected excess returns and variances.
    - V_T : the final wealth in the portfolio
    - P_T : a certain target wealth level at the horizon T.
    - $\mathbf{P}[V_T - P_T \geq 0]$ Browne (1996) and Das et al. (2018) as an objective for maximization by an active portfolio management.
      - This probability is the same as the price of a binary option on the terminal wealth V_T with strike P_T : $\mathbf{P}[V_T - P_T \geq 0] = \mathbb{E}_t\left[\mathbb{1}_{V_T > P_T}\right]$

Introduction : Goal based Wealth management

- Goal-based wealth management
  - this approach uses the price of this binary option as the objective function.
  - This idea can also be modified by using a call option-like expectation $\mathbb{E}_t\left[(V_T - P_T)_+\right]$
    - Such an expectation quantifies how much the terminal wealth is expected to exceed the target, rather than simply providing the probability of such event

Introduction : reference

- This treatment of the goal-based utility function
  - implemented in a reinforcement learning (RL) framework for discrete-time planning problems.
  - RL does not require specific functional forms of the utility
  - nor does it require that the dynamics of the assets be treated as log-normal.
  - RL can be viewed as a data driven extension of dynamic programming (Sutton and Barto, 2018).
  - a substantial challenge with the RL framework is the curse of dimensionality
    - portfolio allocation as a continuous action space Markov Decision Process (MDP) requires techniques such as deep Q learning
      - cumbersome, highly data intensive, heuristics for operational efficiency.(Dixon et al., 2020).
    - other function approximation methods combined e.g. with the Least Squares Policy Iteration (LSPI) method (Lagoudakis and Parr, 2003).
      - exponential complexity with increasing stocks in the portfolio

Introduction : presentation

- present G-learning (Fox et al., 2015) : a probabilistic extension of Q-learning which scales to high dimensional portfolios while providing a flexible choice of utility functions.
    - To demonstrate the utility of G-learning, a general class of wealth management problems:
        - optimization of a defined contribution retirement plan, where cash is injected (rather than withdrawn) at each time step.
        - adopt a more “RL-native" approach by directly specifying one-step rewards.
        - Such an approach is sufficiently general to capture other possible settings, such as e.g. a retirement plan in a decumulation (post-retirement) phase, or target based wealth management.
    - Previously, G learning was applied to dynamic portfolio optimization in (Halperin and Feldshteyn, 2018), while here we extend this approach to portfolio management involving cash ows at intermediate time steps.

Introduction : key step

- A key step in our formulation
  - define actions as absolute (dollar-valued) changes of asset positions
  - transformation of the optimization problem into an unconstrained optimization problem
  - provides a semi-analytical solution for a particular choice of the reward function.
- As will be shown below, this approach offers a tractable setting
  - the direct reinforcement learning problem of learning the optimal policy which maximizes the total reward,
  - its inverse problem where we observe actions of a financial agent but not the rewards by the agent.
    - Inference of the reward function from observations of states and actions of the agent
    - the objective of Inverse Reinforcement Learning (IRL).
  - introduce GIRL (G-learning IRL) a framework for inference of rewards of financial agents that are “implied" by their observed behavior.
- The two practical algorithms, G-Learner and GIRL : wealth management and robo-advising.

G-learning

- overview of G-learning
  - as a probabilistic extension of the popular Q-learning method in reinforcement learning.
- short informal summary of the differences
  - Q-learning : off-policy RL method with a deterministic policy.
  - G-Learning : off-policy RL method with a stochastic policy.
    - G-learning : an entropy-regularized Q-learning
    - suitable when working with noisy data
    - Because G-learning operates with stochastic policies, it amounts to a generative RL model.

Bellman optimality equation

- formally,
  - x_t : a state vector for an agent
    - summarizes the knowledge of the environment that the agent needs in order to perform an action at at time step t.
  - $\hat{R}_t(\mathbf{x}_t, \mathbf{a}_t)$: a random reward collected by the agent for taking action at at time t when the state of the environment is x_t
  - Assume that all future actions at for future time steps are determined according to a policy $\pi(\mathbf{a}_t|\mathbf{x}_t)$
    - which species which action at to take when the environment is in state x_t
  - We note that policy π can be deterministic as in Q-learning, or stochastic as in G-learning

Bellman optimality equation

- For a given policy π , the expected value of cumulative reward with a discount factor γ, conditioned on the current state x_t , defines the value function

$$V_t^{\pi}(\mathbf{x}_t) := \mathbb{E}_t^{\pi}\left[\sum_{t'=t}^{T-1} \gamma^{t'-t}\hat{R}_{t'}(\mathbf{x}_{t'}, \mathbf{a}_{t'})\middle| \mathbf{x}_t\right]$$

- π* : the optimal policy, i.e. the policy that maximizes the total reward.
  - This policy corresponds to the optimal value function, denoted $V_t^{\star}(\mathbf{x}_t)$

$$V_t^{\star}(\mathbf{x}_t) = \max_{\mathbf{a}_t} \hat{R}_t(\mathbf{x}_t, \mathbf{a}_t) + \gamma \mathbb{E}_{t,\mathbf{a}_t}\left[V_{t+1}^{\star}(\mathbf{x}_{t+1})\right]$$

$$\pi_t^{\star}(\mathbf{a}_t|\mathbf{x}_t) = \arg\max_{\mathbf{a}_t} \hat{R}_t(\mathbf{x}_t, \mathbf{a}_t) + \gamma \mathbb{E}_{t,\mathbf{a}_t}\left[V_{t+1}^{\star}(\mathbf{x}_{t+1})\right]$$

- The goal of Reinforcement Learning (RL)
  - solve the Bellman optimality equation based on samples of data

Entropy-regularized Bellman optimality equation

- reformulating the Bellman optimality equation using a Fenchel-type representation:

$$V_t^\star(\mathbf{x}_t) = \max_{\pi(\cdot|y)\in\mathcal{P}} \sum_{\mathbf{a}_t\in\mathcal{A}_t} \pi(\mathbf{a}_t|\mathbf{x}_t)\left(\hat{R}_t(\mathbf{x}_t,\mathbf{a}_t) + \gamma\mathbb{E}_{t,\mathbf{a}_t}\left[V_{t+1}^\star(\mathbf{x}_{t+1})\right]\right)$$

  - $\mathcal{P} = \left\{\pi : \pi \geq 0, \mathbb{1}^T\pi = 1\right\}$: denotes a set of all valid distributions
- The one-step information cost of a learned policy $\pi(\mathbf{a}_t|\mathbf{x}_t)$ relative to a reference policy $\pi_0(\mathbf{a}_t|\mathbf{x}_t)$

$$g^\pi(\mathbf{x}_t,\mathbf{a}_t) := \log\frac{\pi(\mathbf{a}_t|\mathbf{x}_t)}{\pi_0(\mathbf{a}_t|\mathbf{x}_t)}$$

Entropy-regularized Bellman optimality equation

Its expectation with respect to the policy π is the Kullback-Leibler (KL) divergence of $\pi(\cdot|\mathrm{x}_t)$ and $\pi_0(\cdot|\mathrm{x}_t)$:

$$\mathbb{E}_\pi\left[\, g^\pi(\mathrm{x},\mathrm{a})|\,\mathrm{x}_t\right] = KL[\pi||\pi_0](\mathrm{x}_t) := \sum_{\mathrm{a}_t} \pi(\mathrm{a}_t|\mathrm{x}_t)\log\frac{\pi(\mathrm{a}_t|\mathrm{x}_t)}{\pi_0(\mathrm{a}_t|\mathrm{x}_t)}. \tag{6}$$

The total discounted information cost for a trajectory is defined as follows:

$$I^\pi(\mathrm{x}_t) := \sum_{t'=t}^{T} \gamma^{t'-t}\mathbb{E}_t^\pi\left[\, g^\pi(\mathrm{x}_{t'},\mathrm{a}_{t'})|\,\mathrm{x}_t\right]. \tag{7}$$

The *free energy* function $F_t^\pi(\mathrm{x}_t)$ is defined as the value function (4) augmented by the information cost penalty (7) which is added using a regularization parameter $1/\beta$:

$$F_t^\pi(\mathrm{x}_t) := V_t^\pi(\mathrm{x}_t) - \frac{1}{\beta}I^\pi(\mathrm{x}_t) = \sum_{t'=t}^{T}\gamma^{t'-t}\mathbb{E}_t^\pi\left[\hat{R}_{t'}(\mathrm{x}_{t'},\mathrm{a}_{t'}) - \frac{1}{\beta}g^\pi(\mathrm{x}_{t'},\mathrm{a}_{t'})\right]. \tag{8}$$

Entropy-regularized Bellman optimality equation

- The free energy, $F_t^{\pi}(\mathbf{x}_t)$: the entropy-regularized value function
  - the amount of regularization : tuned to the level of noise in the data.
  - The regularization parameter controls a trade-off between reward optimization and proximity of the optimal policy to the reference policy
  - often referred to as the “inverse temperature" parameter, using the analogy between Eq.(8) and free energy in physics, see e.g. (Dixon et al., 2020).
  - The reference policy, π_0, provides a “guiding hand" in the stochastic policy optimization process that we now describe.

$$F_t^\pi(\mathrm{x}_t) := V_t^\pi(\mathrm{x}_t) - \frac{1}{\beta} I^\pi(\mathrm{x}_t) = \sum_{t'=t}^{T} \gamma^{t'-t} \mathbb{E}_t^\pi \left[\hat{R}_{t'}(\mathrm{x}_{t'}, \mathrm{a}_{t'}) - \frac{1}{\beta} g^\pi(\mathrm{x}_{t'}, \mathrm{a}_{t'}) \right]. \quad (8)$$

$$g^\pi(\mathrm{x}_t, \mathrm{a}_t) := \log \frac{\pi(\mathrm{a}_t|\mathrm{x}_t)}{\pi_0(\mathrm{a}_t|\mathrm{x}_t)}. \quad (5)$$

Entropy-regularized Bellman optimality equation

A Bellman equation for the free energy function $F_t^\pi(\mathrm{x}_t)$ is obtained from Eq.(8):

$$F_t^\pi(\mathrm{x}_t) = \mathbb{E}_{\mathrm{a}|y} \left[\hat{R}_t(\mathrm{x}_t, \mathrm{a}_t) - \frac{1}{\beta} g^\pi(\mathrm{x}_t, \mathrm{a}_t) + \gamma \mathbb{E}_{t,\mathrm{a}} \left[F_{t+1}^\pi(\mathrm{x}_{t+1}) \right] \right]. \quad (9)$$

For a finite-horizon setting with a terminal reward $\hat{R}_T(\mathrm{x}_t, \mathrm{a}_T)$, Eq.(9) should be supplemented by a terminal condition

$$F_T^\pi(\mathrm{x}_t) = \hat{R}_T(\mathrm{x}_t, \mathrm{a}_T^\star) \quad (10)$$

where the final action $\mathrm{a}_T^\star$ maximizes the terminal reward $\hat{R}_T$ for the given terminal state x_T. Eq.(9) can be viewed as a soft probabilistic relaxation of the Bellman equation for the value function, with the KL information cost penalty (5) as a regularization controlled by the inverse temperature β. In addition to such a regularized value function (free energy), we will next introduce an entropy regularized Q-function.

$$F_t^\pi(\mathrm{x}_t) := V_t^\pi(\mathrm{x}_t) - \frac{1}{\beta} I^\pi(\mathrm{x}_t) = \sum_{t'=t}^{T} \gamma^{t'-t} \mathbb{E}_t^\pi \left[\hat{R}_{t'}(\mathrm{x}_{t'}, \mathrm{a}_{t'}) - \frac{1}{\beta} g^\pi(\mathrm{x}_{t'}, \mathrm{a}_{t'}) \right]. \quad (8)$$

G-function: an entropy-regularized Q-function

- the state-action free energy function $G^\pi(x,a)$

$$\begin{aligned} G_t^\pi(\mathrm{x}_t, \mathrm{a}_t) &= \hat{R}_t(\mathrm{x}_t, \mathrm{a}_t) + \gamma \mathbb{E}\left[F_{t+1}^\pi(\mathrm{x}_{t+1}) \middle| \mathrm{x}_t, \mathrm{a}_t \right] \\ &= \hat{R}_t(\mathrm{x}_t, \mathrm{a}_t) + \gamma \mathbb{E}_{t,\mathrm{a}} \left[\sum_{t'=t+1}^{T} \gamma^{t'-t-1} \left(\hat{R}_{t'}(\mathrm{x}_{t'}, \mathrm{a}_{t'}) - \frac{1}{\beta} g^\pi(\mathrm{x}_{t'}, \mathrm{a}_{t'}) \right) \right] \\ &= \mathbb{E}_{t,\mathrm{a}_t} \left[\sum_{t'=t}^{T} \gamma^{t'-t} \left(\hat{R}_{t'}(\mathrm{x}_{t'}, \mathrm{a}_{t'}) - \frac{1}{\beta} g^\pi(\mathrm{x}_{t'}, \mathrm{a}_{t'}) \right) \right], \end{aligned}$$

- compare this expression with Eq.(8), we obtain the relation between the G-function and the free energy F

$$F_t^\pi(\mathrm{x}_t) = \sum_{\mathrm{a}_t} \pi(\mathrm{a}_t|\mathrm{x}_t) \left[G_t^\pi(\mathrm{x}_t, \mathrm{a}_t) - \frac{1}{\beta} \log \frac{\pi(\mathrm{a}_t|\mathrm{x}_t)}{\pi_0(\mathrm{a}_t|\mathrm{x}_t)} \right]. \quad (12)$$

This functional is maximized by the following distribution $\pi(\mathrm{a}_t|\mathrm{x}_t)$:

$$\pi(\mathrm{a}_t|\mathrm{x}_t) = \frac{1}{Z_t} \pi_0(\mathrm{a}_t|\mathrm{x}_t) e^{\beta G_t^\pi(\mathrm{x}_t, \mathrm{a}_t)} \quad (13)$$

$$Z_t = \sum_{\mathrm{a}_t} \pi_0(\mathrm{a}_t|\mathrm{x}_t) e^{\beta G_t^\pi(\mathrm{x}_t, \mathrm{a}_t)}.$$

$$F_t^{\pi}(\mathrm{x}_t) = \sum_{\mathrm{a}_t} \pi(\mathrm{a}_t|\mathrm{x}_t)\left[G_t^{\pi}(\mathrm{x}_t,\mathrm{a}_t) - \frac{1}{\beta}\log\frac{\pi(\mathrm{a}_t|\mathrm{x}_t)}{\pi_0(\mathrm{a}_t|\mathrm{x}_t)}\right]. \quad (12)$$

$$\pi(\mathrm{a}_t|\mathrm{x}_t) = \frac{1}{Z_t}\pi_0(\mathrm{a}_t|\mathrm{x}_t)e^{\beta G_t^{\pi}(\mathrm{x}_t,\mathrm{a}_t)}$$

$$Z_t = \sum_{\mathrm{a}_t} \pi_0(\mathrm{a}_t|\mathrm{x}_t)e^{\beta G_t^{\pi}(\mathrm{x}_t,\mathrm{a}_t)}.$$

G-function: an entropy-regularized Q-function

- the optimal solution

$$F_t^{\pi}(\mathrm{x}_t) = \frac{1}{\beta}\log Z_t = \frac{1}{\beta}\log\sum_{\mathrm{a}_t}\pi_0(\mathrm{a}_t|\mathrm{x}_t)e^{\beta G_t^{\pi}(\mathrm{x}_t,\mathrm{a}_t)}$$

- the optimal action policy

$$\pi(\mathrm{a}_t|\mathrm{x}_t) = \pi_0(\mathrm{a}_t|\mathrm{x}_t)e^{\beta\left(G_t^{\pi}(\mathrm{x}_t,\mathrm{a}_t) - F_t^{\pi}(\mathrm{x}_t)\right)}$$

$$G_t^{\pi}(\mathrm{x}_t,\mathrm{a}_t) = \hat{R}_t(\mathrm{x}_t,\mathrm{a}_t) + \gamma\mathbb{E}_{t,\mathrm{a}}\left[F_{t+1}^{\pi}(\mathrm{x}_{t+1})\middle|\mathrm{x}_t,\mathrm{a}_t\right]$$

$$G_T^{\pi}(\mathrm{x}_t,\mathrm{a}_T^{\star}) = \hat{R}_T(\mathrm{x}_t,\mathrm{a}_T^{\star})$$
$$F_T^{\pi}(\mathrm{x}_t) = G_T^{\pi}(\mathrm{x}_t,\mathrm{a}_T^{\star}) = \hat{R}_T(\mathrm{x}_t,\mathrm{a}_T^{\star})$$

G-learning

- $$G_t^{\pi}(\mathrm{x}, \mathrm{a}) = \hat{R}(\mathrm{x}_t, \mathrm{a}_t) + \mathbb{E}_{t,\mathrm{a}} \left[\frac{\gamma}{\beta} \log \sum_{\mathrm{a}_{t+1}} \pi_0(\mathrm{a}_{t+1}|\mathrm{x}_{t+1}) e^{\beta G_{t+1}^{\pi}(\mathrm{x}_{t+1}, \mathrm{a}_{t+1})} \right]. \qquad (18)$$
    - a soft relaxation of the Bellman optimality equation for the action-value Q-function
    - The "inverse-temperature" parameter determines the strength of entropy regularization
        - take a zero-temperature as $\beta \to \infty$
        - Because the last term in (18) approximates the max(.) function when β is large but finite, uniform reference distribution π_0, is known in the literature as soft Q-learning"
    - For finite values $\beta < \infty$, in a setting of Reinforcement Learning with observed rewards, Eq.(18) can be used to specify G-learning (Fox et al., 2015): an off-policy time-difference (TD) algorithm that generalizes Q-learning to noisy environments where an entropy-based regularization appropriate.

G-learning

- The G-learning algorithm of Fox et al. (2015)
  - a tabulated setting where both the state and action space are finite
- In our case, we model MDPs in high-dimensional continuous state and action spaces
  - cannot rely on a tabulated G-learning
  - need to specify a functional form of the action-value function
  - use a non-parametric function approximation such as a neural network to represent its values
  - An additional challenge is to compute a multidimensional integral (or a sum) over all next-step actions in Eq.(18)
    - repeated numerical integration of this integral can substantially slow down the learning.

G-learning

- To summarize
  - G-learning
    - off-policy, generative reinforcement learning algorithm with a stochastic policy
- In the next section
  - an approach to goal-based wealth management based on G-learning
- Later in this paper,
  - G-learning for Inverse Reinforcement Learning (IRL)

Portfolio optimization for a defined contribution retirement plan

- considering a simplified model for retirement planning
  - assume a discrete-time process with T steps
    - T is the (integer-valued) time horizon
  - The investor/planner keeps the wealth in N assets
    - x_t : the vector of dollar values of positions in different assets at time t
      - the first asset with n = 1 : a risk-free bond
      - other assets : risky, with uncertain returns r_t whose expected values are $\bar{\mathbf{r}}_t$
      - The covariance matrix of return is Σ_r of size (N-1)x(N-1)
    - u_t : the vector of changes in these positions
- Optimization of a retirement plan :
  - optimization of both regular contributions to the plan and asset allocations
  - The pair (c_t, u_t) : considered the action variables in a dynamic optimization problem corresponding

    c_t : a cash installment in the plan at time t.

Portfolio optimization for a defined contribution retirement plan

- $\hat{P}_{t+1}$: a pre-specified target value of a portfolio at time t + 1 at each time step t
- target value $\hat{P}_{t+1}$ at step t exceeds the next-step value $V_{t+1} = (1+r_t)(x_t + u_t)$ of the portfolio
  - seek to impose a penalty for under-performance relative to this target
  - To this end, we can consider the following expected reward for time step t

$$R_t(\mathbf{x}_t, \mathbf{u}_t, c_t) = -c_t - \lambda \mathbb{E}_t \left[\left(\hat{P}_{t+1} - (1 + \mathbf{r}_t)(\mathbf{x}_t + \mathbf{u}_t) \right)_- \right] - \mathbf{u}_t^T \mathbf{\Omega} \mathbf{u}_t . \qquad (19)$$

    - first term : an installment of amount c_t at the beginning of time period t
    - second term : the expected negative reward from the end of the period for under-performance
    - the third term : approximates transaction costs by a convex functional with the parameter matrix and serves as a L2 regularization

$$R_t(\mathbf{x}_t, \mathbf{u}_t, c_t) = -c_t - \lambda \mathbb{E}_t \left[\left(\hat{P}_{t+1} - (1 + \mathbf{r}_t)(\mathbf{x}_t + \mathbf{u}_t) \right)_+ \right] - \mathbf{u}_t^T \mathbf{\Omega} \mathbf{u}_t . \quad (19)$$

Portfolio optimization for a defined contribution retirement plan

- The one-step reward (19) : inconvenient to work with due to non-linearity $()_+$:=max(.,0) under the expectation
- Another problem : decision variables c_t and u_t are not independent $$\sum_{n=1}^{N} u_{tn} = c_t, \quad (20)$$
  - at every time step, the total change in all positions should equal the cash installment c_t at this time.
- modify the one-step reward (19) in two ways
  - replace the first term using Eq.(20), and approximate the rectified non-linearity by a quadratic function

$$R_t(\mathbf{x}_t, \mathbf{u}_t) = -\sum_{n=1}^{N} u_{tn} - \lambda \mathbb{E}_t \left[\left(\hat{P}_{t+1} - (1 + \mathbf{r}_t)(\mathbf{x}_t + \mathbf{u}_t) \right)^2 \right] - \mathbf{u}_t^T \mathbf{\Omega} \mathbf{u}_t$$

    - explicitly resolves (20) between the cash injection ct and portfolio allocation decisions
      - converts the initial constrained optimization problem into an unconstrained one
    - quadratic in actions u_t : therefore highly tractable

$$R_t(\mathbf{x}_t, \mathbf{u}_t, c_t) = -c_t - \lambda \mathbb{E}_t \left[\left(\hat{P}_{t+1} - (1 + \mathbf{r}_t)(\mathbf{x}_t + \mathbf{u}_t) \right)_- \right] - \mathbf{u}_t^T \mathbf{\Omega} \mathbf{u}_t . \quad (19)$$

Portfolio optimization for a defined contribution retirement plan

- the well known disadvantage of quadratic rewards (penalties)
  - symmetric, and penalize both scenarios $V_{t+1} \gg \hat{P}_{t+1}$ and $V_{t+1} \ll \hat{P}_{t+1}$
    - while in fact we only want to penalize the second class of scenarios
  - To mitigate this drawback
    - target values $\hat{P}_{t+1}$: considerably higher than the time-t expectation of the next-period portfolio value
    - set the target portfolio as a linear combination of a portfolio-independent benchmark B_t and the current portfolio growing with a fixed rate η :

$$\hat{P}_{t+1} = (1 - \rho) B_t + \rho \eta \, \mathbf{1}^{\mathbf{T}} \mathbf{x}_{\mathbf{t}}, \quad (22)$$

$$R_t(\mathbf{x}_t, \mathbf{u}_t) = -\sum_{n=1}^{N} u_{tn} - \lambda \mathbb{E}_t \left[\left(\hat{P}_{t+1} - (1 + \mathbf{r}_t)(\mathbf{x}_t + \mathbf{u}_t) \right)^2 \right] - \mathbf{u}_t^T \mathbf{\Omega} \mathbf{u}_t. \quad (21)$$

Portfolio optimization for a defined contribution retirement plan

- The expected reward (21) : more explicit quadratic form
  - asset returns : $\mathbf{r}_t = \bar{\mathbf{r}}_t + \tilde{\varepsilon}_t$
    - the first component $\bar{r}_0(t) = r_f$: the risk-free rate
    - $\tilde{\varepsilon}_t = (0, \varepsilon_t)$: idiosyncratic noise with Σ_r of size (N-1)x(N-1)

$$\begin{aligned} R_t(\mathbf{x}_t, \mathbf{u}_t) &= -\lambda \hat{P}_{t+1}^2 - \mathbf{u}_t^T \mathbb{1} + 2\lambda \hat{P}_{t+1} (\mathbf{x}_t + \mathbf{u}_t)^{\mathrm{T}} (\mathbf{1} + \bar{\mathbf{r}}_t) - \lambda (\mathbf{x}_t + \mathbf{u}_t)^{\mathrm{T}} \hat{\mathbf{\Sigma}}_t (\mathbf{x}_t + \mathbf{u}_t) - \mathbf{u}_t^{\mathrm{T}} \mathbf{\Omega} \mathbf{u}_t \\ &= \mathbf{x}_t^T \mathbf{R}_t^{(xx)} \mathbf{x}_t + \mathbf{u}_t^T \mathbf{R}_t^{(ux)} \mathbf{x}_t + \mathbf{u}_t^T \mathbf{R}_t^{(uu)} \mathbf{u}_t + \mathbf{x}_t^T \mathbf{R}_t^{(x)} + \mathbf{u}_t^T \mathbf{R}_t^{(u)} + R_t^{(0)} \end{aligned}$$

$$\hat{\mathbf{\Sigma}}_t = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_r \end{bmatrix} + (1 + \bar{\mathbf{r}}_t)(1 + \bar{\mathbf{r}}_t)^T$$

$$\mathbf{R}_t^{(xx)} = -\lambda \eta^2 \rho^2 \mathbf{1}\mathbf{1}^{\mathrm{T}} + \mathbf{2}\lambda \eta \rho (\mathbf{1} + \bar{\mathbf{r}}_t)\mathbf{1}^{\mathrm{T}} - \lambda \hat{\mathbf{\Sigma}}_t$$

$$\mathbf{R}_t^{(ux)} = 2\lambda \eta \rho (1 + \bar{\mathbf{r}}_t)\mathbf{1}^T - 2\lambda \hat{\mathbf{\Sigma}}_t$$

$$\mathbf{R}_t^{(uu)} = -\lambda \hat{\mathbf{\Sigma}}_t - \mathbf{\Omega}$$

$$\mathbf{R}_t^{(x)} = -2\lambda \eta \rho (1 - \rho) B_t \mathbf{1} + 2\lambda (1 - \rho) B_t (1 + \bar{\mathbf{r}}_t)$$

$$\mathbf{R}_t^{(u)} = -\mathbf{1} + 2\lambda (1 - \rho) B_t (1 + \bar{\mathbf{r}}_t)$$

$$R_t^{(0)} = -(1 - \rho)^2 \lambda B_t^2$$

G-learner for retirement plan optimization

- use a semi-analytical formulation of G-learning with Gaussian time-varying policies (GTVP)
  - the G-Learner algorithm
    - We start by specifying a functional form of the value function as a quadratic form of x_t

$$F_t^{\pi}(\mathbf{x}_t) = \mathbf{x}_t^T \mathbf{F}_t^{(xx)} \mathbf{x}_t + \mathbf{x}_t^T \mathbf{F}_t^{(x)} + F_t^{(0)}, \tag{24}$$

$$\mathbf{x}_{t+1} = \mathbf{A}_t (\mathbf{x}_t + \mathbf{u}_t) + (\mathbf{x}_t + \mathbf{u}_t) \circ \tilde{\varepsilon}_t, \quad \mathbf{A}_t := \mathrm{diag}(1 + \bar{\mathbf{r}}_t), \quad \tilde{\varepsilon}_t := (0, \varepsilon_t) \tag{25}$$

$$F_t^{\pi}(\mathbf{x}_t) = \frac{1}{\beta} \log \int \pi_0(\mathbf{u}_t | \mathbf{x}_t) e^{\beta G_t^{\pi}(\mathbf{x}_t, \mathbf{u}_t)} d\mathbf{u}_t$$

$$\pi_0(\mathbf{u}_t | \mathbf{x}_t) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_p|}} e^{-\frac{1}{2}(\mathbf{u}_t - \hat{\mathbf{u}}_t)^T \Sigma_p^{-1} (\mathbf{u}_t - \hat{\mathbf{u}}_t)},$$

GIRL: G-learning IRL

- So far in this paper
  - considered the setting of (direct) reinforcement learning
    - the agent (investor) learns while observing the rewards,
    - optimizes the policy so that the expected cumulative reward (regularized by the KL information cost) is maximized
    - This setting is suitable when the investor explicitly defines his or her reward function.
- individual investor : not be able to explain his or her utility function used for trading decision-making
  - an agent (investor) : behavioral inference to a different agent (a researcher or robo-advisor)
  - robo-advisor has access to observed trajectories (states and actions) of the agent
    - not to rewards received by the agent
  - Such cases where rewards not available belong in the realms of Inverse Reinforcement Learning (IRL) whose objective is to recover both the reward function of the agent and the optimal policy, see e.g. (Dixon et al., 2020) for a review.

GIRL: G-learning IRL

- the IRL problem with G-learning : GIRL (G-learning IRL)
  - inference of the reward function of an individual agent such as a retirement plan contributor or an individual brokerage account holder
  - given a history of dollar-nominated asset positions in an investment portfolio with an agent's decisions
    - both injections or withdrawals of cash from the portfolio
    - asset allocation decisions
    - Additionally, historical values of asset prices and expected asset returns for all assets in the investor universe

GIRL: G-learning IRL

- historical data that includes a set of D trajectories ζ_i
  - i = 1,... D of state-action pairs (x_t u_t) where trajectory i starts at some time t_{0i} and runs until time T_i .
- Consider a single trajectory ζ from this collection
  - set for this trajectory the start time t = 0 and the end time T
  - As individual trajectories are considered independent, they will enter additively in the final log-likelihood of the problem
  - assume that dynamics are Markovian in the pair (x_t u_t), with a generative model

$$p_\theta(\mathbf{x}_{t+1}, \mathbf{u}_t|\mathbf{x}_t) = \pi_\theta(\mathbf{u}_t|\mathbf{x}_t) p_\theta(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$$

- for a vector of model parameters, and is the action policy given by Eq.(38).
- The probability of observing trajectory

$$P(\mathbf{x}, \mathbf{u}|\Theta) = p_0(\mathbf{x}_0) \prod_{t=0}^{T-1} \pi_\theta(\mathbf{u}_t|\mathbf{x}_t) p_\theta(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$$

GIRL: G-learning IRL

-