UNIVERSITY OF TWENTE.

The unified framework for sequential decisions

Anne Zander

Sources

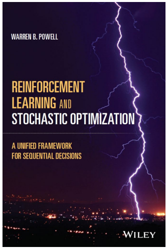

WARREN B. POWELL
REINFORCEMENT LEARNING AND STOCHASTIC OPTIMIZATION
A UNIFIED FRAMEWORK FOR SEQUENTIAL DECISIONS
WILEY

Figure: Reinforcement Learning and Stochastic Optimization: A unified framework for sequential decisions, `https://ut.on.worldcat.org/oclc/1304342551`

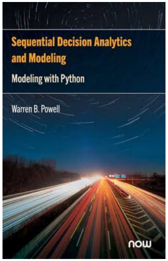


Figure: Sequential Decision Analytics and Modeling, `https://castle.princeton.edu/wp-content/uploads/2022/11/Powell-SDAM-Nov242022_final_w_frontcover.pdf`

Learning goals

Be able to

- model a sequential decision problem using the unified framework for sequential decisions,
- assess the applicability of the four meta-policies to solve a sequential decision problem,
- solve a sequential decision problem by evaluating and tuning a policy.

Overview

Examples of sequential decision-making problems

- Transportation: Which vehicle should serve which request?
- Supply chain management: When should we place orders to stock our inventory?
- Health sciences: Which drug designs to test next?
- Health care logistics: When to schedule a surgery?
- Business: Which products to sell at what price?
- Finance: When to buy or sell an asset?
- Energy: When to charge and discharge a battery connected to the energy market?
- Robotics: What (directional) force to apply to move a robot?

The jungle of stochastic optimization

Stochastic programming
Robust optimization
Decision analysis
Approximate dynamic programming
Simulation optimization
Optimal learning
Model predictive control
Dynamic Programming and control
Stochastic search
Bandit problems
Optimal control
Reinforcement learning
Online computation
Stochastic control
Markov decision processes
Simulation optimization

Introduction to Stochastic Programming
Handbook of Simulation Optimization
Robust Optimization
Introduction to Decision Analysis
Approximate Dynamic Programming
Optimal Learning
Multi-Armed Bandit Allocation Indices
Dynamic Programming and Optimal Control
Model Predictive Control
Optimal Control
Introduction to Stochastic Search and Optimization
Active Learning
Reinforcement Learning
Markov Decision Processes
Online Computation and Competitive Analysis
Stochastic Simulation Optimization

Overview

A universal canonical model

Model first, then solve!

- State: $S_t \in \mathcal{S}$
- Decision and policy: $x_t = X_t^\pi(S_t)$ with $x_t \in \mathcal{X}_t(S_t)$
- Exogenous information: W_{t+1}
- Transition function: $S_{t+1} = S^M(S_t, x_t, W_{t+1})$
- Contribution/Reward: $C_t(S_t, x_t, W_{t+1})$
- (Finite horizon cumulative reward) objective function: $\max_\pi \mathbb{E}\left\{\sum_{t=0}^{T-1} C_t(S_t, X_t^\pi(S_t), W_{t+1})\right\}$

Example sequential decision problem

- Managing the inventory of a product:
  We need to order some quantity of a product and store it in inventory (both incurs costs). Then, the demand for the product becomes known and is satisfied if possible.

The state variable 1

The state

is a function of history which, combined with a decision (the policy) and the exogenous information, contains all the information that is necessary and sufficient to model our system from time *t* onward.

The state variable 2

State $=$

- Physical state: R_t
  - Inventory: Quantity of product in stock
- Information state: I_t
  - Prices for purchasing, storing and selling of the product (if they change over time)
- Belief state: B_t captures beliefs about unknown parameters or quantities
  - How the market prices might change in the future

The state variable 3

- (Dynamic) state $S_t = (R_t, I_t, B_t)$: Only contains information that changes over time
- Latent variables: Not part of the state variable; Contains (deterministic) parameters which do not change over time

- Dynamic:
  - Inventory
- Latent variables:
  - Cost for ordering one product (if it does not change over time)

The decision

In general, a decision can be

- An element of a discrete set ($x \in \{x_1, \ldots, x_M\}$)
- A continuous vector ($x \in \mathbb{R}^n$)
- An integer vector ($x \in \mathbb{Z}^n$)
- $\ldots$

with potentially additional constraints

- How much product should be ordered? Non-negative integer, potentially with an upper bound

The exogenous information

W_{t+1} is the information that becomes known after making decision x_t at time t

- After ordering a certain quantity of the product, we observe the demand for that product.

The transition function

Given the state and decision at time t, as well as the exogenous information, calculate the next state:

$$S_{t+1} = S^M(S_t, x_t, W_{t+1})$$

- New product quantity stored = max (Old product quantity stored + ordered product quantity - demand, 0)

The contribution

The contribution $C_t(S_t, x_t, W_{t+1})$ at time t is an immediate reward w.r.t. the state, decision and potentially also the exogenous information.

- $-$ costs for ordering product $-$ costs for holding product $+$ revenue from fulfilling demand

The objective function

The objective function ties together the contributions over time into on single measure, which we want to optimize. Here, we use the finite horizon cumulative reward objective function:

$$\max_{\pi} \mathbb{E} \left\{ \sum_{t=0}^{T-1} C_t(S_t, X_t^{\pi}(S_t), W_{t+1}) \right\}.$$

Modeling and solution process

Step 1: Narrative description of the problem.

Step 2: Identify core elements without mathematics:

- What metrics are we trying to impact?
- What decisions are being made?
- What are the different sources of uncertainty?

Step 3: Model in the universal framework

a In words

b Mathematical model

Step 4: The uncertainty model

Step 5: Designing policies

Step 6: Evaluating policies

Managing inventory, Step 1

- Order a certain quantity of product at the end of the day (costs)
- Ordered product will be delivered the next day
- Demand for product becomes known on that next day
- Demand is satisfied using stored and newly delivered product (revenue)
- Remaining product will be stored (costs)

Managing inventory, Step 2

- What metrics are we trying to impact?

Managing inventory, Step 2

- What metrics are we trying to impact?
    - Profit
- What decisions are being made?

Managing inventory, Step 2

- What metrics are we trying to impact?
    - Profit
- What decisions are being made?
    - Ordering of products (which quantity)
- What are the different sources of uncertainty?

Managing inventory, Step 2

- What metrics are we trying to impact?
  - Profit
- What decisions are being made?
  - Ordering of products (which quantity)
- What are the different sources of uncertainty?
  - Demand for the product

Managing inventory, Step 3a

- State: Quantity of product in inventory
- Decision: How much product to order
- Exogenous information: Demand for product
- Transition function: New product quantity stored = max (Old product quantity stored + ordered product quantity - demand, 0)
- Contribution/Rewards: $-$ costs for ordering $-$ costs for holding inventory $+$ revenue for fulfilling demand

Exercise: Sequential decision problem

- Discuss with your neighbor and come up with a sequential decision problem.
- Do steps 1, 2 and 3a together.

Managing inventory, Step 3a

- State: Quantity of product in inventory
- Decision: How much product to order
- Exogenous information: Demand for product
- Transition function: New product quantity stored = max (Old product quantity stored + ordered product quantity - demand, 0)
- Contribution/Rewards: $-$ costs for ordering $-$ costs for holding inventory $+$ revenue for fulfilling demand

Managing inventory, Step 3b

- State:
  - Dynamic: $R_t \in \mathbb{N}_0$ inventory of product (at the end of day t)
  - Latent variables:
    - c_p: costs for purchasing one quantity
    - c_h: costs for holding one quantity for one day
    - p: price for selling one quantity

Managing inventory, Step 3b

- State:
  - Dynamic: $R_t \in \mathbb{N}_0$ inventory of product (at the end of day t)
  - Latent variables:
    - c_p: costs for purchasing one quantity
    - c_h: costs for holding one quantity for one day
    - p: price for selling one quantity
- Decision: $x_t \in \mathbb{N}_0$: Number of quantities ordered (at the end of day t) and delivered at the beginning of day $t + 1$

Managing inventory, Step 3b

- State:
  - Dynamic: $R_t \in \mathbb{N}_0$ inventory of product (at the end of day t)
  - Latent variables:
    - c_p: costs for purchasing one quantity
    - c_h: costs for holding one quantity for one day
    - p: price for selling one quantity
- Decision: $x_t \in \mathbb{N}_0$: Number of quantities ordered (at the end of day t) and delivered at the beginning of day $t + 1$
- Exogenous information: W_{t+1} is the realized demand $\hat{D}_{t+1} \in \mathbb{N}_0$ over day $t + 1$

Managing inventory, Step 3b

- State:
  - Dynamic: $R_t \in \mathbb{N}_0$ inventory of product (at the end of day t)
  - Latent variables:
    - c_p: costs for purchasing one quantity
    - c_h: costs for holding one quantity for one day
    - p: price for selling one quantity
- Decision: $x_t \in \mathbb{N}_0$: Number of quantities ordered (at the end of day t) and delivered at the beginning of day $t+1$
- Exogenous information: W_{t+1} is the realized demand $\hat{D}_{t+1} \in \mathbb{N}_0$ over day $t+1$
- Transition function: $R_{t+1} = \max\{R_t + x_t - \hat{D}_{t+1}, 0\}$

Managing inventory, Step 3b

- State:
  - Dynamic: $R_t \in \mathbb{N}_0$ inventory of product (at the end of day t)
  - Latent variables:
    - c_p: costs for purchasing one quantity
    - c_h: costs for holding one quantity for one day
    - p: price for selling one quantity
- Decision: $x_t \in \mathbb{N}_0$: Number of quantities ordered (at the end of day t) and delivered at the beginning of day $t+1$
- Exogenous information: W_{t+1} is the realized demand $\hat{D}_{t+1} \in \mathbb{N}_0$ over day $t+1$
- Transition function: $R_{t+1} = \max\{R_t + x_t - \hat{D}_{t+1}, 0\}$
- Contribution: $C(R_t, x_t, \hat{D}_{t+1}) = -c_p x_t - R_t c_h + p \min\{\hat{D}_{t+1}, R_t + x_t\}$
- Objective: $\max_\pi \mathbb{E}\left\{\sum_{t=0}^{T-1} C(R_t, X_t^\pi(R_t), \hat{D}_{t+1})\right\}$

Managing inventory, Step 4

The uncertainty model:

- Assume that $\hat{D}_{t+1}$ is the realization of random variable D_{t+1} which is Poisson distributed with parameter $\bar{D}$ for every t
- Add $\bar{D}$ to the latent variables

Overview

Solving a sequential decision problem

Challenge: Find a policy that solves

$$\max_{\pi} \mathbb{E} \left\{ \sum_{t=0}^{T-1} C_t(S_t, X_t^{\pi}(S_t), W_{t+1}) \right\}$$

Knowledge of the model

Model: Transition function + contribution function + distribution of the exogenous information

We distinguish between problems where:

- Model-based: The model is known, i.e., the transition function, the contribution function and the distribution of the exogenous information are known
- Model-free: The model is unknown, i.e., either the transition function, the contribution function or the exogenous information is unknown (or any combination thereof)

Solving to optimality

Model-based:

- Model sequential decision problem (finite state and decision sets) as a stochastic dynamic program or Markov decision process:
    - Transition function and exogenous information are merged into transition probabilities
    - $P(S_{t+1}|S_t, x_t) = \sum_w P(W_{t+1} = w)\mathbb{1}_{S_{t+1}=S^M(S_t,x_t,w)}$
- Apply dynamic programming methods:
    - Backward recursion
    - Value iteration, policy iteration, linear programming

Model-free:

- (Tabular) reinforcement learning

Extensions inventory problem

- Several products need to be stored in a warehouse with limited capacity
- Extra cost for ordering independent of the quantity order
- Cost reduction for ordering large quantities
- Longer time lags between ordering and delivery
- Uncertainty with respect costs and prices
- ...

Dynamic programming curses

Three curses of dimensionality:

- State space,
- Decision space,
- Outcome space: Refers to the number of possible next states

Curse of modeling: Calculation of transition probabilities

- Descriptive model: Knowledge of transition probabilities
- Generative model: Knowledge of transition function and distribution of exogenous information

Overview

Two fundamental strategies to design policies

A policy is a rule that determines a feasible decision given the available information in a state.

- Policy search: Search within a family of (parameterized) functions to find the best via tuning and comparing.
- Lookahead approximations: Construct policies by approximating the impact of a decision now on the future.

Results in four meta-classes of policies.

Overview

Policy function approximations (PFAs)

Policy function approximations (PFAs) are functions that map states to decisions. They can be

- lookup tables,
- parametric functions,
- or nonparametric (continuous) functions.

Example: $X^{PFA}(S_t \mid \theta) = \theta_0 + \theta_1 \phi_1 (S_t) + \theta_2 \phi_2 (S_t) + \dots$

- PFAs are typically limited to discrete actions, or low-dimensional (and typically continuous) vectors
- Time dependency, state dependency, and decision dimensionality limit PFAs' usefulness

Example inventory problem

- State:
  - Dynamic: $R_t \in \mathbb{N}_0$ inventory of product (at the end of day t)
  - Latent variables:
    - c_p: costs for purchasing one quantity
    - c_h: costs for holding one quantity for one day
    - p: price for selling one quantity
- Decision: $x_t \in \mathbb{N}_0$: Number of quantities ordered (at the end of day t) and delivered at the beginning of day $t+1$
- Exogenous information: $W_{t+1} = \hat{D}_{t+1}$ is the realized demand $\hat{D}_{t+1} \in \mathbb{N}_0$ over day $t+1$
- Transition function: $R_{t+1} = \max\{R_t + x_t - \hat{D}_{t+1}, 0\}$
- Contribution:
  $C(R_t, x_t, \hat{D}_{t+1}) = -c_p x_t - R_t c_h + p \min\{\hat{D}_{t+1}, R_t + x_t\}$
- Objective: $\max_\pi \mathbb{E}\left\{\sum_{t=0}^{T-1} C(R_t, X_t^\pi(R_t), \hat{D}_{t+1})\right\}$

Example inventory problem

Order-up-to policy:

$$X^{PFA}(R_t \mid \theta) = \begin{cases} \theta^{\max} - R_t & \text{if } R_t < \theta^{\min} \\ 0 & \text{otherwise} \end{cases}$$

where $\theta = \left(\theta^{\min}, \theta^{\max}\right)$ is a set of parameters that needs to be determined.

Cost function approximations (CFAs)

Cost function approximations (CFAs) are parameterized optimization models:

- parameterized modification of the contribution function
- subject to (possibly parameterized) approximation of constraints (set of feasible decisions)

$$X_t^{CFA}(S_t|\theta) = \arg \max_{x \in \mathcal{X}_t^{CFA}(\theta)} \mathbb{E}\{C_t^{CFA}(S_t, x, W_{t+1}|\theta)\}$$

- Well-suited for complex, high dimensional problems (e.g., airline scheduling)

Example stochastic traveling salesman

Traveling salesman problem:
Find an order of n cities to visit such that the total travel time is minimized.

A stochastic version:
The travel time from city i to j is known after traveling from i to j and is a realization of a random variable with parameters dependent on i, j and the realized travel time between the node before i and i.

A CFA-policy:
Drive to city with lowest $\theta\%$-quantile of the travel time.

Policy search – tuning

Both PFAs and CFAs result in having to solve:

$$\max_{\theta} F^{PFA/CFA}(\theta) = \max_{\theta} \mathbb{E}\left\{\sum_{t=0}^{T-1} C_t(S_t, X^{PFA/CFA}(S_t|\theta), W_{t+1})\right\}$$

Overview

Lookahead approximations

Lookahead approximations take decisions based on approximating the future expected sum of contributions given a decision and assuming (close to) optimal future decisions.

Objective: $\max_{\pi} \mathbb{E} \left\{ \sum_{t=0}^{T-1} C_t(S_t, X_t^{\pi}(S_t), W_{t+1}) \right\}$

Optimal decision at time t:

$$X_t^* (S_t)$$
$$= \arg\max_{x_t} \left(\mathbb{E} \left\{ C_t \left(S_t, x_t, W_{t+1} \right) \right\} + \max_{\pi} \mathbb{E} \left\{ \sum_{t'=t+1}^{T-1} C_t \left(S_{t'}, X_{t'}^{\pi} \left(S_{t'} \right), W_{t'+1} \right) \right\} \right)$$

Value function approximations (VFAs)

Value function at time *t* and Bellman equation:

$$\begin{aligned} & V_t\left(S_t\right) \\ = & \max_{x \in \mathcal{X}_t(S_t)} \left(\mathbb{E}\left\{C_t\left(S_t, x, W_{t+1}\right)\right\} + \max_\pi \mathbb{E}\left\{ \sum_{t'=t+1}^{T-1} C_{t'}\left(S_{t'}, X_{t'}^\pi\left(S_{t'}\right), W_{t'+1}\right) \right\} \right) \\ = & \max_{x \in \mathcal{X}_t(S_t)} \left(\mathbb{E}\left\{C_t\left(S_t, x, W_{t+1}\right)\right\} + \mathbb{E}\left\{V_{t+1}(S_{t+1})\right\}\right) \end{aligned}$$

Value function approximations (VFAs)

Value function approximations (VFAs) are policies based on approximating the value of being in a state.

$$\begin{aligned} X_t^{VFA}(S_t|\theta) &= \arg\max_{x \in \mathcal{X}_t(S_t)} \left(\mathbb{E}\left\{ C_t(S_t, x, W_{t+1}) \right\} + \mathbb{E}\left\{ \bar{V}_{t+1}(S_{t+1}|\theta) \right\} \right) \\ \bar{V}_{t+1}(S_{t+1}|\theta) &\approx \max_{\pi} \mathbb{E}\left\{ \sum_{t'=t+1}^{T-1} C_{t'}(S_{t'}, X_{t'}^{\pi}(S_{t'}), W_{t'+1}) \right\} \end{aligned}$$

Use VFAs for problems, where

- we need to capture the impact of a decision on the future,
- the value can be captured in a well-defined function (i.e., having structural properties we can exploit).

Value function approximations (VFAs)

- **VFAs assuming a descriptive model**: Approximate dynamic programming working with the state-value function V
- **VFAs assuming a generative model or no model**: Reinforcement learning working with the state-action-value function Q

Example state aggregation

Consider a SDP with a known (descriptive) model but with a large number of possible states. Instead of determining a value for each state separately, we could apply state aggregation:

- Group several states together with one representative state each
- Do backward recursion with the representative states only:
  - Start with $\bar{V}_T(Z_T) = 0$ for all possible representative states Z_T at time T
  - For $t = T-1, \ldots, 0$ and each possible rep. state Z_t compute $\bar{V}_t(Z_t) = \max_{x \in \mathcal{X}_t(Z_t)} \left(\mathbb{E}\left\{ C_t\left(Z_t, x, W_{t+1}\right)\right\} + \mathbb{E}\left\{ \bar{V}_{t+1}(Z_{t+1})\right\}\right)$
  - $X_t^{VFA}(S_t) = \arg\max_{x \in \mathcal{X}_t(S_t)} \left(\mathbb{E}\left\{ C_t(S_t, x, W_{t+1})\right\} + \mathbb{E}\left\{ \bar{V}_{t+1}(Z_{t+1})\right\}\right)$

Direct lookahead approximations (DLAs)

Direct lookahead approximations (DLAs) directly approximate the downstream value.

$$X^{DLA}(S_t) = \arg\max_{x \in \mathcal{X}_t(S_t)} \Bigg(\mathbb{E}\left\{C_t(S_t, x, W_{t+1})\right\}]$$

$$+ \text{ Approximation of } \max_{\pi} \mathbb{E}\left\{ \sum_{t'=t+1}^{T-1} C_{t'}\left(S_{t'}, X^{\pi}_{t'}\left(S_{t'}\right), W_{t'+1}\right) \right\} \Bigg)$$

Strategies:

- ▶ Limit the horizon,
- ▶ Use a deterministic approximation of the future,
- ▶ Use a sample of random outcomes,
- ▶ Use simple future policies, i.e., rollout policies.

- ▶ As uncertainty increases, VFAs tend to work better than DLAs

Example chess program

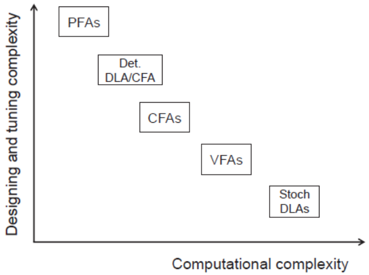
A chess program may:

- Try a next move A and simulate the rest of the game several times using simple rollout policies to determine the propotion of won games.
- Do the same with move B,C, et.
- Make the move with the highest proportion of won games.

Overview

Choosing the policy class (1)

Work your way up from the (computationally) simplest policy class.



The price of simplicity is tunable parameters, and tuning is hard!

Hybrid policies are also possible.

Choosing the policy class (2)

Screening questions and suggested starting strategies:

- Does the problem have structure that suggests a simple and natural decision? PFA
- Would a greedy (i.e., myopic) policy work reasonably well? Likely CFA. Look for parameterizations to improve performance.
- Is the problem fairly stationary, or highly non-stationary? A deterministic DLA (with imbedded forecast) can turn a non-stationary problem into a stationary one.

Choosing the policy class (3)

- Will decisions that you might make in the future affect what you are going to do now? Deterministic DLA and stochastic DLA (when uncertainty becomes relevant); If you have significant uncertainty, DLAs start to struggle, and VFAs become more attractive.
- Does the value of the most important state variables appear to have natural structure that can be exploited in the design of a value function approximation?
  VFA

For more complex problems, be prepared to use a hybrid policies.

Choosing the policy class (4)

Consider the knowledge of the model, e.g.,

- CFAs need the knowledge of the contribution function
- DLAs need the complete model (or a good approximation thereof)

Consider computation power and computation time constraints, e.g., how much computation time/power do you have available

- Before you have to start taking decisions in the real world?
- Inbetween decision epochs in the real world?

Exercise asset selling

Asset selling problem: We are holding a share of stock, looking for an opportune time to sell within a time horizon of size T. If we sell at time t, we receive a price that varies according to some random process over time. Once we sell the stock, the process stops.

Design several reasonable policies considering the mathematical model on the next slide!

Exercise asset selling

- State: $S_t = (R_t, p_t)$ with
  $R_t = \begin{cases} 1 & \text{if we are holding the stock at time } t, \\ 0 & \text{if we are no longer holding the stock at time } t \end{cases}$
  and p_t the price per share of stock at time t
- Decision: $x_t = \begin{cases} 1 & \text{if we sell the stock at time } t \\ 0 & \text{if do not sell the stock at time } t \end{cases}$ with $x_t \leq R_t$
- Exogenous information: $W_{t+1} = \hat{\epsilon}_{t+1}$ change in price of a share of stock with $\hat{\epsilon}_{t+1}$ the realization of a $\mathcal{N}(0, \sigma^2)$ RV
- Transition function: $R_{t+1} = R_t - x_t$ and $p_{t+1} = p_t + \hat{\epsilon}_{t+1}$
- Contribution: $C((R_t, p_t), x_t) = x_t p_t$

Overview

Summary

- Universal modeling framework
- Example: Inventory management
- 4 meta policies: PFA, CFA, VFA, DLA
- Strategies to find a fitting meta policy

Next

- How to evaluate a given policy?
- How to tune policies?

UNIVERSITY OF TWENTE.

Policy evaluation and tuning

Anne Zander

Overview

Policy evaluation

For a given policy X^π, we want to compute:

$$F^\pi = \mathbb{E}\left\{\sum_{t=0}^{T-1} C_t(S_t, X_t^\pi(S_t), W_{t+1})\right\}$$

Stochastic modeling of the exogenous information

- Descriptive model: Use transition probabilities
- Generative model: Numerical sampling of ex. information
- No model:
  - Historical data on ex. information
  - Observational sampling of ex. information

Assume a generative model in the following.

Basic notation for information processes

A sample path $\omega \in \Omega$ refers to complete sequence of exogenous information with $\Omega =$ set of all possible sample paths.

For example (assuming W_{t+1} to be independent of S_t and x_t):

Sample path	$t = 1$	$t = 2$	$\dots$	$t = T$
ω^1	$W_1\left(\omega^1\right)$	$W_2\left(\omega^1\right)$	$\dots$	$W_T\left(\omega^1\right)$
ω^2	$W_1\left(\omega^2\right)$	$W_2\left(\omega^2\right)$	$\dots$	$W_T\left(\omega^2\right)$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
ω^K	$W_1\left(\omega^K\right)$	$W_2\left(\omega^K\right)$	$\dots$	$W_T\left(\omega^K\right)$

Policy evaluation

Given X^π and a sample path ω, we compute:

$$\hat{F}^\pi(\omega) = \sum_{t=0}^{T-1} C_t\Big(S_t(\omega), X_t^\pi(S_t(\omega)), W_{t+1}(\omega)\Big)$$

using $S_{t+1}(\omega) = S^M(S_t(\omega), X_t^\pi(S_t(\omega)), W_{t+1}(\omega))$.

Let $\hat{\Omega} = \{\omega^1, \ldots, \omega^K\}$ and compute an approximation to F^π as:

$$\bar{F}^\pi = \frac{1}{K} \sum_{k=1}^{K} \hat{F}^\pi(\omega^k)$$

Model asset selling

- State: $S_t = (R_t, p_t)$ with
  $$R_t = \begin{cases} 1 & \text{if we are holding the stock at time } t, \\ 0 & \text{if we are no longer holding the stock at time } t \end{cases}$$
  and p_t the price per share of stock at time t
- Decision: $x_t = \begin{cases} 1 & \text{if we sell the stock at time } t \\ 0 & \text{if do not sell the stock at time } t \end{cases}$ with $x_t \leq R_t$
- Exogenous information: $W_{t+1} = \hat{\epsilon}_{t+1}$ change in price of a share of stock with $\hat{\epsilon}_{t+1}$ the realization of a $\mathcal{N}(0, \sigma^2)$ RV
- Transition function: $R_{t+1} = R_t - x_t$ and $p_{t+1} = p_t + \hat{\epsilon}_{t+1}$
- Contribution: $C((R_t, p_t), x_t) = x_t p_t$

Example asset selling problem

Write down pseudocode for evaluating the following policy in the asset selling problem:

$$X^{\text{high-low}}\left((R_t, p_t) \mid \theta^{\text{high-low}}\right) = \begin{cases} 1 & \text{if } p_t < \theta^{low} \text{ or } p_t > \theta^{\text{high}} \text{ and } R_t = 1 \\ 1 & \text{if } t = T - 1 \text{ and } R_t = 1 \\ 0 & \text{otherwise} \end{cases}$$

where we assume fixed values of $\theta^{\text{high-low}} = (\theta^{\text{high}}, \theta^{\text{low}})$.

Pseudocode policy evaluation

```
Input: $\theta^{\text{high}}, \theta^{\text{low}}, S_0 = (R_0, p_0), \sigma^2, K, T$ with $R_0 = 1$
  $Fbar = 0$
  $Fhat = zeroarray(K)$
  **for** $k = 0, \dots, K-1$ **do**
      $price = p_0$
      $assest = R_0$
      **for** $t = 0, \dots, T-1$ **do**
          **if** $price < \theta^{\text{low}}$ or $price > \theta^{\text{high}}$ or $t = T-1$ and $asset = 1$ **then**
              $Fhat(k) = price$
              BREAK
          **end if**
          $price = price +$ draw.normal$(0, \sigma^2)$
      **end for**
  **end for**
  **for** $k = 0, \dots, K-1$ **do**
      $Fbar = Fbar + \frac{1}{K} Fhat(k)$
  **end for**
Output: $Fbar$
```

Overview

Stochastic search problem

Find best (non-time-dependent) PFA/CFA-parameters = solve stochastic search problem:

$$\max_{\theta} \mathbb{E} F(\theta, W) = \max_{\theta} \mathbb{E} \left\{ \sum_{t=0}^{T-1} C_t \left(S_t, X_t^{\pi} \left(S_t \mid \theta \right), W_{t+1} \right) \right\}$$

Stochastic search solution strategies:

- Analytical/special structure
- Sampled models
- Adaptive approach: SDP over different θs, apply the four meta-policies

Sampled models for parameter tuning

Sampled Model (Sample Average Approximation):

Pick a set of sample paths $\hat{\Omega} = \{\omega^1, \ldots, \omega^K\}$ and solve

$$\begin{aligned} &\arg\max_\theta \bar{F}^\pi(\theta) \\ =&\arg\max_\theta \frac{1}{K}\sum_{n=1}^{K} \hat{F}^\pi(\theta|\omega^n) \\ =&\arg\max_\theta \frac{1}{K}\sum_{n=1}^{K} \left\{ \sum_{t=0}^{T-1} C_t\left(S_t(\omega^n), X_t^\pi\left(S_t(\omega^n) \mid \theta\right), W_{t+1}(\omega^n)\right) \right\} \end{aligned}$$

Parameter tuning

Parameter tuning needs to consider the following issues:

- Lab vs. field experiments
- Tunable parameters: *The price of simplicity is tunable parameters, and tuning is hard!*
- Latent variables: if they change, we usually need to retune
- Expensive experiments

Parameter tuning is a sequential decision problem to solve a sequential decision problem.

A weak search algorithm can produce a poor policy.

Exercise

Work on the exercise about evaluating and tuning policies for the asset selling problem

Overview

Summary

- Policy evaluation
- Policy tuning

Outlook

- The state can not be observed completely: Partial observability, POMDPs
- Continuous control
- Several decision-makers: Multiagent systems
- Adversarial transitions/worst-case analysis: Online algorithms
- Infinite horizon problems: average reward and cumulative discounted reward objective functions
- Several reward signals: Multi-objective problems
- Risk-advers/seeking decision-makers $\rightarrow$ different objective functions
- ....

Exercise policy evaluation and tuning

1. Tuning for the asset selling problem

Use the Python project Exercise.zip. In the project there are three scripts and an Excel file, where you can adjust parameter values. If you run 'AssetSellingDriverScript.py', you either evaluate the asset selling policy high_low several times for fixed parameter values or you perform a grid search for those parameters.

In the Excel file, you can adjust the parameters of the model (ignore all other parameters):

- high_low param1 in Sheet1: θ^{low}
- high_low param2 in Sheet1: θ^{high}
- low_min in Sheet2: Lowest value for θ^{low} in the grid search
- low_max in Sheet2: Highest value for θ^{low} in the grid search
- high_min in Sheet2: Lowest value for θ^{high} in the grid search
- high_max in Sheet2: Highest value for θ^{high} in the grid search
- increment_size in Sheet2: Discretization increment for both θ^{low} and θ^{high} in the grid search
- Policy in Sheet3: Can be "high_low" or "full_grid"
- TimeHorizon in Sheet3: Time horizon T of the asset selling SDP
- InitialPrice in Sheet3: Initial price p_0
- Iterations in Sheet3: Number of sample paths

In case of using the parameter value "high_low" for Policy, in the console for each time step and iteration, you will see the decision and the current price. Then, per iteration, you will see the contribution and the cumulative average contribution (average of all previous contributions).

In case of using the parameter value "full_grid" for Policy, in the console for each time step, iteration, and combination of parameter values, you will see the decision and the current price. In the end, you will see the cumulative average contribution per iteration (averaged over all contributions of all previous iterations) per combination of parameter values.

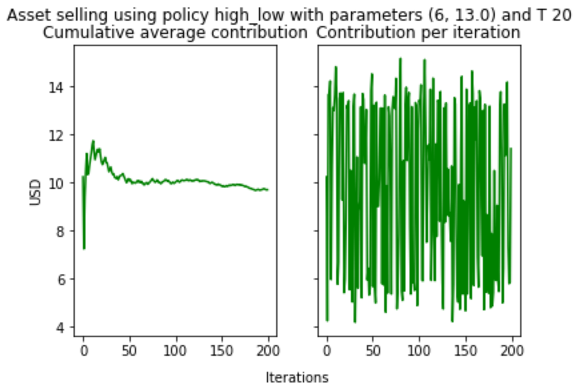
- Evaluate the high_low policy with parameter values $param1 = 6$ and $param2 = 13$ and initial price 10 over a time horizon of 20 using 200 iterations. Leave all other parameters as they are. Report and interpret the results. Here, also make use of the generated plot.

- Perform a grid search for the high_low policy with parameter values low_min= 8, low_max= 9, high_min= 11, high_max= 13, and increment_size= 1. Leave all other parameters as they are. Report and interpret the results. Here, also make use of the generated plot. Based on the results, which parameter values for the high_low policy would you chose?

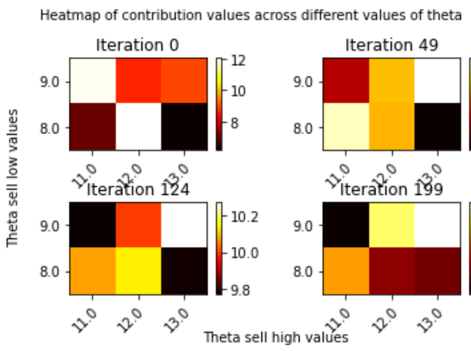
This question is based on question 2.8 in SDAM (`https://castle.princeton.edu/wp-content/uploads/2022/11/Powell-SDAM-Nov242022_final_w_frontcover.pdf`) and on an adjusted version of the code provided by `https://github.com/wbpowell328/stochastic-optimization`.

Solution:

1. (1 pt) The output of the console gives the contribution and the cumulative average contribution per iteration when we apply our fixed high-low-policy. This output is also visualized in the generated plot (contribution per iteration on the right and cumulative average contribution on the left). We see that there is quite some variability in the contribution which is due to the price uncertainty. However, after 200 iterations the average contribution seems to stabilize around 9.7, which is smaller than the initial prize of 10 (therefore the policy which sells immediately would (very likely) be better).



2. (1 pt) The generated plot shows heat maps of the resulting cumulative average contributions for each combination of parameter values after the first, 50th, 125th and 200th iteration, respectively. The console gives the cumulative average contribution values after the 200 iterations explicitly. Assuming that after 200 iterations the resulting average contributions are significantly different, we would choose the combination the gives the highest average contribution, which is ≈ 10.29 for $\theta^{low} = 9$ and $\theta^{high} = 13$, which is higher than the initial prize of 10.


Heatmap of contribution values across different values of theta
Iteration 0
Iteration 49
Iteration 124
Iteration 199
Theta sell low values
Theta sell high values
9.0
8.0
11.0
12.0
13.0
12
10
8
10.0
9.8
10.2
10.0
9.8
10.2
10.0