# Introduction to Markov Decision Processes

Martin L. Puterman and Timothy C. Y. Chan

July 20, 2023

# Chapter 1

# Current status

1. to write (Divide book into 3 sections: Foundations, Classic MDP, RL Overview)
2. revisit for footnotes, optimality criteria
3. revisit for footnotes -make sure optimal stopping agrees with 6 in formulation and notation, clinical decision making example reformulate.
4. revisit for footnotes
5. Marty clinical decision making example, update figures, publish
6. Tim to print, read and edit; MLP add state-action, take out parking.
7. work through Nicolas Gast feedback
8. Delete - keep until done
9. Tim to reread one-period model section in light of changes (decision rule primitive). Tim to fix algorithms. Marty to revise machine preventive maintenance example and figure.
10. Add weighted supremum norms - Marty reread for wordings.
11. Reorganize the chapter: preliminaries, episodic, discounted. Within episodic and discounted we'd have Monte Carlo and then TD. Marty to edit. What data structure are we using. add ucb sampling (probably only works for bandits?) Episodic = transient. Divide chapter into Episodic, Discounted and Average?. Move Stochastic Approximation to Appendix. Add problems.
12. Structure: Parallel 11 as much as possible Episodic vs. discounted, first evaluation then optimization, add policy gradients. ?Larger example? Pritam

Appendices Marty read LP, Tim to read MC part; move regression to book appendix.

Should we add something on bandits? code to Github

# Chapter 2

# Examples and Applications



***We welcome all feedback and suggestions at:***
***martin.puterman@sauder.ubc.ca and tcychan@mie.utoronto.ca***

*For the things we have to learn before we can do them, we learn by doing them.*
*Aristotle, Philosopher, 384-322 BC*

Representing a dynamic decision problem as a Markov decision process requires specifying all of the model components described previously: decision epochs, states, actions, transition probabilities and rewards. In this chapter, we show how to do so by identifying these objects in many different examples. We have chosen applications from a broad range of disciplines to illustrate both the wide applicability of the Markov decision process formalism and the many features of model formulation. We encourage the reader to attempt formulating the models before seeing how we did so. We conclude the chapter with a section that provides guidance on how to formulate Markov decision process models. The problems at the end of the chapter provide further opportunities to formulate Markov decision processes or revise the examples presented herein when assumptions differ.

## 2.1 Revenue Management: Using Price to Manage Demand

This problem describes a different approach to inventory management. It pertains primarily to products that decline in value over time. As a concrete example, consider the challenges faced by our acquaintance, Frank Z., the former owner of a chain of women's fashion stores in Vancouver, Canada. He related to us that "Setting prices is like a game of chance; if I mark down prices too early in the season, I lose revenue, but if I wait too late, I've lost the opportunity to sell my inventory and also incur costs to store it."

Let us formalize Frank's problem. A retailer has an inventory of $M$ units of a product at the beginning of the season and requires a policy to vary prices over the $N$ month season so as to maximize revenue. We assume prices are set at the beginning of each month, to be chosen from a finite set of prices, and cannot be changed during the month. Assume that when the price is $a$ the demand in period $n$ is random and Poisson distributed with rate $\lambda_{n,a}$. It is reasonable to assume that $\lambda_{n,a}$ is non-increasing in $n$ because fashion products may become less trendy as time goes on and expected demand will decrease. It is also reasonable to assume that $\lambda_{n,a}$ is non-increasing in $a$ since in a typical demand curve the quantity demanded decreases as price increases.

Assume a monthly holding cost of $h(s)$ when the end of month inventory equals $s$ units with $h(0) = 0$. Any goods left over at the end of month $N$ are sold to an outlet store at a low price of $H$ per unit, representing the scrap value.

**Decision Epochs:** Prices are set at the beginning of each month, so

$$T = \{1, 2, \ldots, N\}.$$

**States:** States represent the number of items in stock at the start of each month in the planning horizon:

$$S = \{0, 1, \ldots, M\}.$$

**Actions:** Actions represent the price to set in each period. Assuming there are are $K$ candidate prices for all $s \in S$,

$$A_s = \{a_1, a_2, \ldots, a_K\}.$$

We assume that $a_1 \leq a_2 \leq \cdots \leq a_K$ and that $a_1 = H$ denotes the scrap value.

**Rewards:** If the inventory at the end of the month is $j$, that means $s - j \geq 0$ units were sold during that month. Thus, for $n < N$, $a_k \in A_s$, and $s \in S$,

$$r_n(s, a_k, j) = \begin{cases} a_k(s - j) - h(j), & j = 0, \ldots, s \\ 0, & j = s + 1, \ldots \end{cases}$$

and $r_N(s) = a_1 s$.

**Transition Probabilities:** Since no inventory is added during the planning horizon, only transitions to states with ending inventory $j \leq s$ have nonzero probability. If the demand is at most $s-1$, then the ending inventory is at least 1. If the demand equals or exceeds $s$, then the ending inventory is 0. This logic leads to the following transition probabilities:

$$p_n(j|s,a) = \begin{cases} e^{-\lambda_{n,a}} \lambda_{n,a}^{s-j}/(s-j)! & j = 1, \ldots, s \\ \sum_{i=s}^{\infty} e^{-\lambda_{n,a}} \lambda_{n,a}^{i}/i! & j = 0 \\ 0 & j = s+1, \ldots \end{cases}$$

**Application Challenges**

Applying this model presents two challenges: determining the set of mark down prices and estimating the time varying demand function parameters. In retail, the mark down prices can be set as a percentage of the original price such as "50% off" or "80% off". Estimating the demand function requires considerable amounts of data and may be product specific. Data from similar products may provide guidance when there is not enough historical data for the product. The parameter $\lambda_{n,a}$ may itself be represented as a function of $n$ and $a$ and learned in the decision problem.

## 2.2 A Periodic Review Inventory Model

*"When you have an inventory-based business, most people think only about the first order," Mr. Green said. With long lead times from the factory in China, he was almost immediately trying to figure out how big his next order should be. Underestimating would hurt not just his sales but the status of his Amazon listing; overestimating would drain cash upfront, and he would incur further charges from Amazon for storing excess inventory in its warehouses.*

*The Great Amazon Flip-a-Thon; John Herrman, New York Times, S1,S7, April 4, 2021*

Inventory models represent some of the earliest and most widely studied Markov decision process models. They concern determining appropriate inventory levels for a retail product in the face of future random arrivals of customer orders. As the quote above alludes to, having too little inventory results in losing sales and reputation, while having too much inventory leads to excess storage and capital charges. These costs are key components of inventory models, and this trade-off is the main tension that one needs to balance.

We assume that an *inventory manager* periodically (hourly, daily, weekly or monthly) observes the inventory level of a product and, if deemed opportunistic, places an order

with a *supplier*. The order from a supplier may arrive immediately, by the next review period or after several periods. The delay between placing and receiving an order is referred to as a *lead time*. An inventory model may contain some or all of the following features:

1. **Ordering costs** may consist of a fixed and a variable component. The fixed component $K$ represents the administrative cost of placing an order and the variable component $c(u)$ represents the cost of ordering $u$ units. When no order is placed, the ordering cost is 0.

2. **Holding costs**, denoted by $h(u)$, represent the cost to the inventory manager of storing $u$ units of product for one period when $u > 0$. It is convenient to assume that $h(0) = 0$.

3. **Lost sales** occur if there is insufficient inventory to fulfill demand in the current period. Alternatively, demand may be *backlogged*, meaning that it is not lost and will be fulfilled when future inventory arrives.

4. **Penalty costs**, denoted by $p(u)$, represent the cost to the inventory manager when demand exceeds inventory by $u$ units in one period (i.e., due to backlogging). Assume $p(0) = 0$.

5. **Revenue** of $R(u)$ is received when $u$ units are sold.

6. **Customer demand** for units arrives during a period. The demand distribution may be known or unknown, and may be static or time-varying. Let random variable $D_n$ denote the demand in period $n$. Therefore, the probability that $d$ units are demanded in period $n$ is $P(D_n = d)$.

7. **Product** may have an "infinite" shelf life or may be *perishable*. Perishable items may last one period (for example a newspaper, a loaf of fresh bread, a defrosted vaccine or a seat at a sporting event) or may last for multiple periods (for example blood products, new electronics or fashion goods). Assume product is only available in whole units.

8. **Scrap value**, denoted by $H(s)$, equals the value of the ending inventory when there are $s$ units on hand and the planning horizon is finite. If $s \geq 0$, $H(s)$ may include any potential holding cost before the inventory is liquidated. If $s < 0$, then $H(s)$ represents a penalty associated with not being able to fulfill the $|s|$ items backlogged (e.g., the loss associated with buying these items at a premium from a back-up supplier, or a loss in goodwill due to lost sales).

Formulating this problem as a Markov decision process requires precisely specifying the timing of events. Figure 2.1 depicts the event sequence for the model formulated below. In a typical period, the decision maker reviews the inventory level at a decision epoch and decides how many units of the product to order from the supplier. The

ordered products arrive before the end of the same period. Demand arrives throughout the period and is fulfilled at the end of the period, using inventory on hand as well as the products that arrive from the current period's order. If demand exceeds inventory, it is backlogged for future fulfillment. In the formulation below, we focus on cost minimization, thus ignoring the revenue from selling the inventory, which is left as Exercise 27.

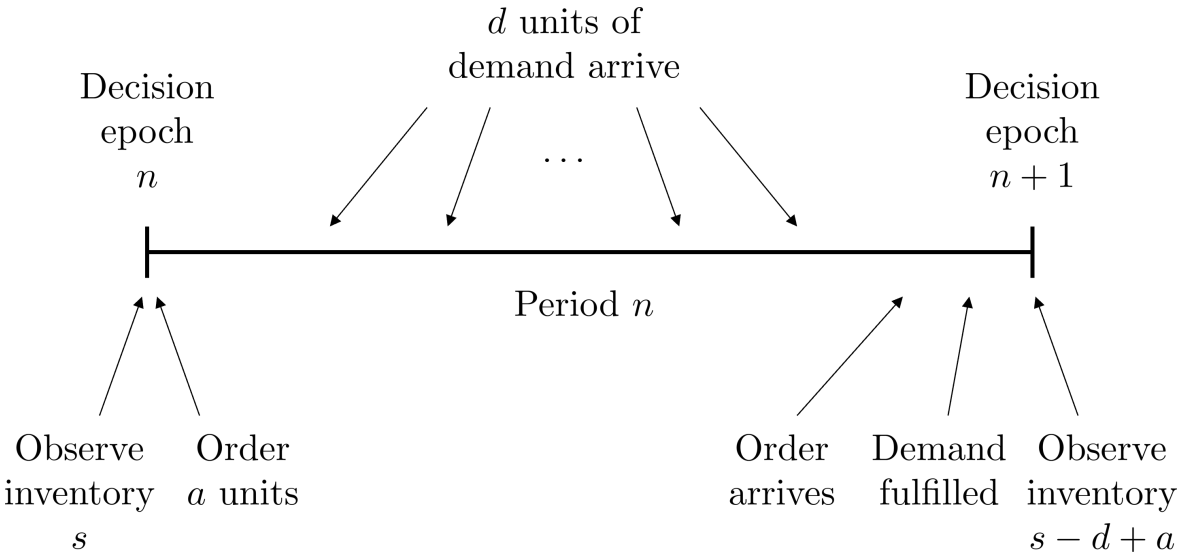


Figure 2.1: Timing of events in the periodic review inventory model described below.

**Decision Epochs:** Decision epochs correspond to the times at which the decision maker reviews the inventory. This problem can be modeled either with a finite or infinite planning horizon. Hence

$$T = \{1, 2, \ldots, N\}, \quad N \leq \infty.$$

**States:** States represent the number of units on hand at a decision epoch. A negative value indicates backlogged demand.

$$S = \{\ldots, -2, -1, 0, 1, 2, \ldots\}.$$

Note that the quantities backlogged and in inventory may be truncated at large values (e.g., the capacity of a warehouse) to ensure a finite state space. Doing so makes the formulation slightly more complex because of the ensuing boundary conditions.

**Actions:** Actions represent the quantity ordered from the supplier for delivery prior to the next decision epoch. For each $s \in S$,

$$A_s = \{0, 1, 2, \ldots\}.$$

Similar to the state space, the action set can be truncated to make it finite.

**Rewards:** Since our formulation assumes reward maximization, we write the reward function as the negative of the costs. The reward consists of the ordering costs, which are incurred if $a > 0$, and holding or penalty costs if $s$ is positive or negative, respectively. Our simplifying assumption is that holding and penalty costs are assessed at the beginning of the period based on the *starting* inventory.

Let $I(\cdot)$ denote an indicator variable. Then the reward can be written

$$r_n(s, a, j) = -KI(a > 0) - c(a) - h(s)I(s > 0) - p(-s)I(s < 0)$$

when $n < N$. When the state at the next decision epoch $n + 1$ is $j$, this means that demand in period $n$ was $s + a - j$. The argument of $p(\cdot)$ is $-s$, which equals the backlogged demand when $s$ is negative. If $N$ is finite, $r_N(s) = H(s)$, the scrap value of the remaining inventory at the end of the planning horizon.

**Transition Probabilities:** Since the state at the next decision epoch cannot exceed $s + a$ (i.e., the demand cannot be negative), the transition probabilities are

$$p_n(j|s, a) = \begin{cases} P(D_n = s + a - j), & j \leq s + a, j \text{ integer} \\ 0, & j = s + a + 1, \ldots \end{cases}$$

### Application Challenges

Applying this model presents several challenges. In particular, one must determine demand distributions, ordering costs, holding costs and penalty costs. Demand distributions may be estimated from historical data; parameterizing the model in terms of a known distribution reduces the challenge in estimating model parameters. Moreover it is likely that demand has seasonal components at the day, week and month levels.

Per unit ordering costs should be easily obtainable but fixed ordering costs may be more challenging to determine. The fixed component encompasses administrative, shipping and handling costs that may be hard to untangle from the per unit cost. Holding costs are real and involve cost of capital and space charges.

Penalty costs are the most challenging to determine since they involve intangibles such as loss of goodwill due to unfulfilled or delayed orders. Instead, the decision maker may specify a service level such as 95% of orders be processed from stock on hand and use a constrained Markov decision process model formulation.

Note that major disruptions to supply chain or consumer behavior arising from, for example, a global pandemic, can lead to significant challenges in estimating appropriate parameters. Procurement costs might be much higher due to increased demand for raw materials and lower manufacturing capacity. Demand for certain products could be significantly increased or decreased compared to historical levels.

## 2.3 Discrete-Time Queuing Models

A queuing system consists of arrivals, a queue and one or more servers. Jobs arrive, wait in a queue if the servers are busy, are served by a free server and then depart the system. Queuing systems have been well-studied in the operations research and engineering literature, and are applicable to a wide variety of service systems, including retail (jobs represent customers), healthcare (jobs represent patients), communication systems (jobs represent packets) and computer systems (jobs represent computing tasks).

From a decision perspective, the most widely studied models are:

1. **Service rate control:** The decision maker varies the service rate to control the queue length and throughput. The service rate may be controlled directly or through the addition and removal of servers.

2. **Admission control:** The decision maker chooses whether or not to admit an arriving job.

3. **Routing control:** In a network of queues, the decision maker chooses how to route jobs based on the workload at each queue.

We will illustrate the formulation of service rate and admission control in the following subsections. A routing control example is provided in Exercise 6.

Some general comments regarding the formulation of discrete-time queuing systems follow:

1. Queuing systems are usually modeled as continuous-time Markov processes or semi-Markov processes. Here, we consider a discrete-time formulation. Assume observation of the system starts at time 0. Let $h$ denote a "small" unit of time and let decision epoch $n$ correspond to "time" $nh$. Let the set of decision epochs be denoted by $\{1, 2, \dots\}$ corresponding to times $\{h, 2h, \dots\}$. Assume $h$ is sufficiently small so that it is very unlikely more than one event (an arrival or service completion) occurs during that time interval.

2. Queuing systems are usually modeled with infinite planning horizons to reflect that they are on-going and decision epochs occur frequently. No terminal reward is specified.

3. Rewards and transition probabilities are assumed to be independent of the decision epoch.

4. The system state is the number of jobs in the queue *and* in service.

### 2.3.1 Service Rate Control

Consider a queuing system with a single server and an infinite capacity queue. Each decision epoch, the decision maker chooses a service "rate" $a_k, k = 1, 2, \ldots, K$ that denotes the probability a job being processed is completed in the current period. Assume the probability a job arrives between two decision epochs is $b$, independent of the number of jobs in the system. We assume that $a_1 \leq a_2 \leq \cdots \leq a_K$ and $a_K + b \leq 1$. The queuing system is shown in Figure 2.2.

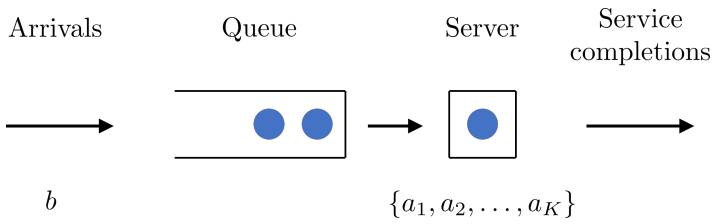


Figure 2.2: Schematic representation of a single server queuing system with adjustable service rate.

Assume a cost $m(a)$ per period for serving at rate $a$, and a delay cost of $f(s)$ per period when there are $s$ jobs in the system, such that both $m(a)$ and $f(s)$ are non-decreasing in their arguments.

**Decision Epochs:**

$$T = \{1, 2, \ldots\}.$$

**States:** States represent the number of jobs in the system (queue plus server):

$$S = \{0, 1, 2, \ldots\}.$$

**Actions:** Actions represent the probability a job currently being served is completed before the next decision epoch. For $s \in S$,

$$A_s = \{a_1, a_2, ...a_K\}.$$

However, given the assumed cost structure, we can assume without loss of generality that $A_0 = \{a_1\}$.

**Rewards:** Costs may be regarded as negative rewards, so

$$r(s, a) = -m(a) - f(s).$$

Note that this reward function is independent of the subsequent state.

**Transition Probabilities:** For $s = 1, 2, \ldots$ and $k = 1, 2, \ldots, K$,

$$p(j|s, a_k) = \begin{cases} a_k & j = s - 1 \\ b & j = s + 1 \\ 1 - a_k - b & j = s. \end{cases} \tag{2.1}$$

For $s = 0$ and $k = 1, 2, \ldots, K$,

$$p(j|0, a_k) = \begin{cases} b & j = 1 \\ 1 - b & j = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

The above formulation assumes an unbounded, discrete state space. For computation, we must truncate the state space at a large value, say $W$, and assume that arrivals are *blocked* when the system state is $W$. This means that to complete the formulation we must add the additional transition probabilities $p(W - 1|W, a_k) = a_k$ and $p(W|W, a_k) = 1 - a_k$ and limit (2.1) to only $s = 1, 2, \ldots, W - 1$.

### 2.3.2 Admission Control

In an admission control model, the decision maker or "gate keeper" decides whether or not to admit an arriving job into a queuing system with fixed arrival and service rates. This example provides an illustration of a model with non-actionable states, which occur when no job arrives in the preceding period.

Assume at most one job can arrive between decision epochs and it does so with probability $b$. If it does not get admitted by the decision maker, the job is lost. Let $h(j)$ be the holding cost when there are $j$ jobs in the system at the start of a period (after the admission decision, but before an arrival or service completion in the same period). Let $w$ be the probability of a service completion between two decision epochs. When a job is admitted to the system, the system receives a payment of $R$. In addition, we assume $b + w \leq 1$, which is reasonable when the time discretization step $h$ is small. Figure 2.3 provides a schematic representation of the system and Figure 2.4 depicts the one-period dynamics.

**Decision Epochs:**

$$T = \{1, 2, \ldots\}.$$

**States:** The state has two components. The first component, denoted $j$, represents the number of jobs in the system and the second component, denoted $k$, indicates whether there is a job waiting for admission ($k = 1$) or not ($k = 0$). Let $J = \{0, 1, \ldots\}$ be the set of possible values $j$. Then the state space is

$$S = J \times \{0, 1\}.$$

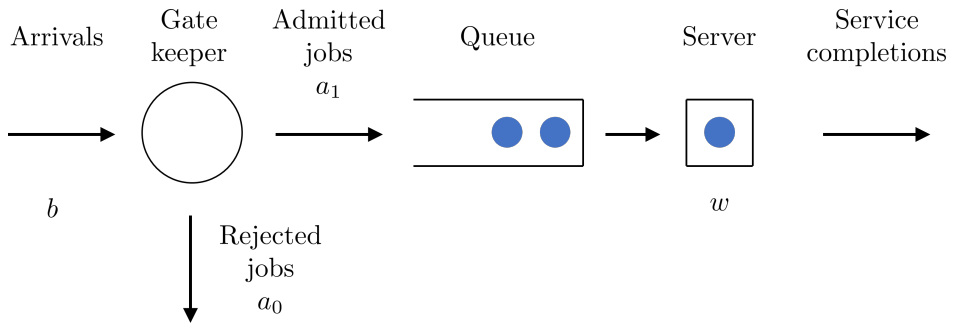


Figure 2.3: Schematic representation of a single server queuing system with admission control.

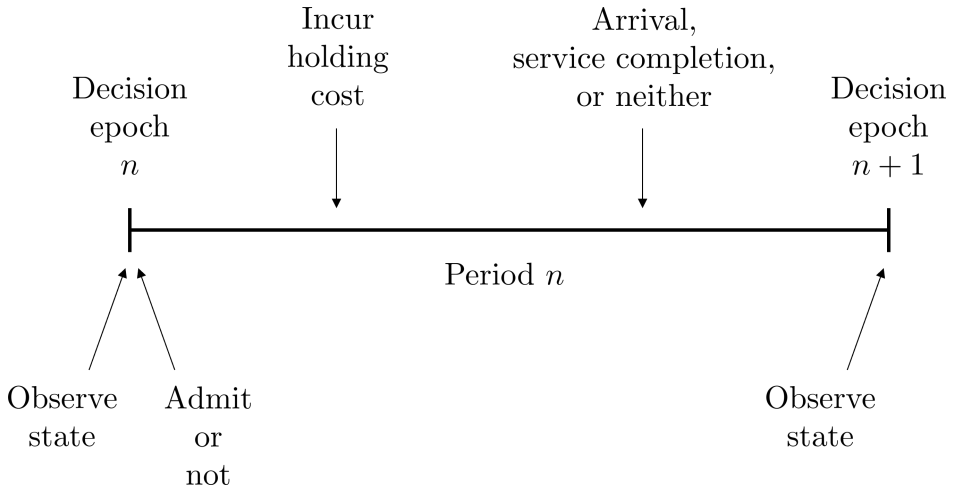


Figure 2.4: Timing of events in the queuing admission control problem.

**Actions:** Let $a_0$ correspond to "do not admit" and $a_1$ be "admit". Since admission is possible only if an arrival occurred since the previous decision epoch, there is no choice in states where $k = 0$. Thus, for any $j = 0, 1, \ldots,$

$$A_{(j,k)} = \begin{cases} \{a_0, a_1\}, & k = 1 \\ \{a_0\}, & k = 0 \end{cases}$$

Recall that to formulate a Markov decision process model, actions need to be specified in all states even when the action set contains a single element.

**Rewards:** For any $j = 0, 1, \ldots,$

$$r((j,k),a) = \begin{cases} R - h(j+1), & k = 1, a = a_1 \\ -h(j), & a = a_0. \end{cases}$$

Note that in this model, the rewards are independent of the subsequent state $(j', k')$.

**Transition Probabilities:** If the action is to not admit ($a = a_0$), then whether there is a job currently waiting to be admitted or not ($k = 0$ or 1) is irrelevant since a waiting job will not be admitted. Thus, for $j = 1, 2, \ldots$, $a = a_0$, and $k = 0$ or 1,

$$p((j', k')|(j, k), a) = \begin{cases} w & j' = j - 1, k' = 0, a = a_0 \\ b & j' = j, k' = 1, a = a_0 \\ 1 - b - w & j' = j, k' = 0, a = a_0. \end{cases} \tag{2.3}$$

Since service completions are not possible if the system is empty, the above dynamics become

$$p((j', k')|(j, k), a) = \begin{cases} b & j' = j, k' = 1, a = a_0 \\ 1 - b & j' = j, k' = 0, a = a_0, \end{cases} \tag{2.4}$$

when $j = 0$, $a = a_0$ and $k = 0$ or 1.

The "admit" action $a_1$ applies only when $k = 1$. So for $j = 0, 1, 2, \ldots$

$$p((j', k')|(j, k), a) = \begin{cases} w & j' = j, k' = 0, a = a_1 \\ b & j' = j + 1, k' = 1, a = a_1 \\ 1 - b - w & j' = j + 1, k' = 0, a = a_1. \end{cases} \tag{2.5}$$

Note the inclusion of $j = 0$ in (2.5). In contrast to (2.3), we can include $j = 0$ into (2.5) since even when the system is empty at the decision epoch, a decision to admit will add a job to the system, which can be completed during the same period with probability $w$.

Figure 2.5 summarizes the possible state transitions for each action. Despite the simplifying assumption that at most one event can happen in a period, keeping track of all transitions requires a careful accounting of events and actions and their interactions. We will revisit this example in Chapter **??**, provide a simpler formulation of the model in terms of the *post-decision state*, and illustrate some computational results. The post-decision state provides an alternative view of the decision timeline depicted in Figure 2.4 that allows the transition dynamics to be modeled more easily. As discussed at the start of the chapter, drawing a correct timeline is an important part of formulating the model correctly. **(June 21 - check that we do this in chapter 5)**.

### Application Challenges

Applying these models requires estimates of arrival probabilities, service probabilities and costs. Queuing models are more commonly formulated in continuous time with inter-arrival and service times modeled using exponential random variables. Thus, if arrivals occur at rate $\lambda$, the probability of one arrival in an interval of length $h$ is given by $\lambda h + o(h)$, the probability of no arrivals in an interval of length $h$ is $1 - \lambda h + o(h)$, and

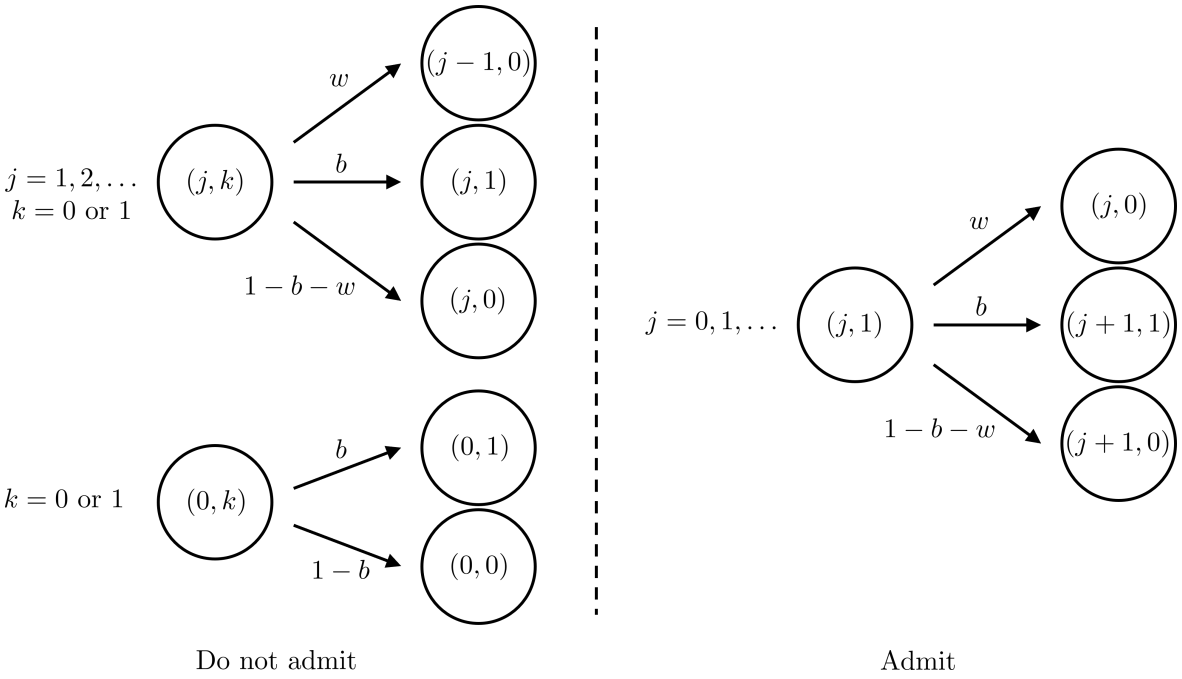


Figure 2.5: Possible state transitions for admission control problem.

the probability of greater than one arrival in an interval of length $h$ is $o(h)$ where $o(h)$ is an expression that converges to zero as $h$ decreases for zero. Thus, it is convenient to set the probability of an arrival in a short time interval of length $h$ to be $\lambda h$.

As in other models, determining costs and rewards is somewhat arbitrary, and may depend heavily on the application. It is important to investigate the impact of specific choices through sensitivity analyses.

## 2.4 Lion Hunting Behavior

Markov decision processes provide a natural framework for modeling behavior when an organism faces a decision that trades off survival with reserving energy. Examples include choosing a location for food acquisition, deciding when to hunt for food, choosing a group size when hunting and deciding when to abandon its offspring. The primary objective in such research is the determine whether an optimization model can explain observed animal behavior.

As an example, consider the challenge facing a lion (*panthera leo*) when deciding to hunt for food. The lion seeks to maximize its probability of survival over a season of $N$ days. A mature lion has an energy storage capacity of $C$ units. Each day it does not hunt the lion depletes its energy reserves by $d$ units. Hunting requires $h$ units of energy with $h > d$. If its energy reserves fall below $c_0$ units, it will not survive to the next day.

At the start of each day, the lion decides whether or not to hunt and if so, what prey to seek. Typically, lions hunt for impalas, gazelles, wildebeests, giraffes and zebras. About half of the time they hunt in groups. Here we assume that the lion hunts alone. (Exercise 20 asks you to formulate the group size decision problem.) Catch probability may vary with species hunted. Assume that there are $M$ species to choose from. Let $w_m$ denote the probability that the lion catches an animal of species $m$; $m = 1, 2, \ldots, M$. We assume a successful hunt for species $m$ yields a total $e_m$ units of energy. For simplicity, we assume all relevant quantities are rounded to the nearest integer.

**Decision Epochs:** Decisions are made at the start of each day during the season, so

$$T = \{1, 2, \ldots, N\}.$$

**States:** The state represents the lion's energy reserves at each decision epoch. Naturally, $S = [0, C]$, but to maintain a finite state formulation, we discretize the state to the nearest energy unit so that

$$S = \{0, 1, \ldots, C\}.$$

**Actions:** Actions in state $s$ may be denoted by

$$A_s = \begin{cases} \{a_0, a_1, \ldots, a_M\} & s \in \{c_0, c_0 + 1, \ldots, C\} \\ \{a_0\} & s \in \{0, 1, \ldots, c_0 - 1\}, \end{cases}$$

where action $a_0$ corresponds to "do not hunt" and action $a_m$ corresponds to "hunt for species $m$".

**Rewards:** Given the lion's survival objective, the lion receives a reward of 1 if it is alive at the end of the season and a reward 0 if it is not. Therefore

$$r_N(s) = \begin{cases} 1 & s \in \{c_0, c_0 + 1, \ldots, C\} \\ 0 & s \in \{0, 1, \ldots, c_0 - 1\}. \end{cases}$$

No rewards are accrued throughout the planning horizon, so $r_n(s, a, j) = 0$ for $n = 1, 2, \ldots, N - 1$, $a \in A_s$, $s \in S$ and $j \in S$.

**Transition Probabilities:** When the lion has energy reserves of $s$ units at the start of the day, hunts for species $m$ and is successful, its energy reserves at the start of the next day is $\min\{s - h + e_m, C\}$. If it is unsuccessful, its energy reserves fall to

$\max\{s-h,0\}$. Therefore

$$p_n(j|s,a) = \begin{cases} 1 & j = \max\{s-d,0\}, s = c_0,\ldots,C, a = a_0 \\ w_m & j = \min\{s-h+e_m, C\}, s = c_0,\ldots,C, a = a_m \\ 1-w_m & j = \max\{s-h,0\}, s = c_0,\ldots,C, a = a_m \\ 1 & j = s, s = 0,1,\ldots,c_0-1, a = a_0 \\ 0 & \text{otherwise.} \end{cases}$$

Note that the transition probabilities take into account the fact that the lion's energy level cannot exceed the maximum capacity or fall below 0, and if its energy level falls below $c_0$, it cannot hunt.

### Application Challenges

This application highlights the fact that it can be challenging to determine parameter values for a Markov decision process, and to do so, one must often appeal to a wide range of sources. Moreover, in this particular example, it is essential to understand the underlying animal behavior and model it correctly, ideally with expert input. An added benefit of developing a formal Markov decision process model is that it identifies relevant parameters that can motivate related research.

The ecology literature suggests values for many of the key model parameters although not always exactly in the form needed. One can use $C = 30$ and $d = 6$ kilograms, based on averages of male and female lions. If a lion-specific parameter is not available from the literature, borrowing values from other species may provide some guidance. For example, wild dogs expend about 23% more energy per hour when hunting, with the average duration of a hunt approximately equal to 40 minutes. Noting that lions undertake on average three chases per day and assuming that they expend the same incremental amount of energy while hunting, on a day they decide to hunt, they will spend 3% more energy than on day they decide to rest so that $h = 1.03d$.

The literature suggests that gazelles yield a mean biomass of 12 kilograms with a catch probability of 0.15 on a single hunt. Observational data suggests that a lion may hunt up to three times a day if earlier hunts are unsuccessful; such dynamics can be incorporated into our model. Modeling the hunting of zebras presents additional challenges. Zebras yield an estimated 164 kilograms of edible biomass with a catch probability between 0.15 and 0.19 depending on the hunt location. Since they are large, their carcasses last for several days and are shared among several lions. Determining how much is available for the hunter requires further assumptions. Other sources provide estimates of the edible biomass for other types of prey such as impalas (29 kg), wildebeests (150 kg), and giraffes (468 kg), but not catch probabilities, which are harder to estimate. It is quite common for domain-specific literature to report data that allows us to estimate only some of the parameters of a Markov decision process, since such data is typically reported without a decision-making objective in mind. As a result, many assumptions must often be made, as described above. Especially

when estimates are quite variable and many assumptions are needed, we recommend conducting sensitivity analyses of these parameters.

## 2.5 Clinical Decision Making: An Application to Liver Transplantation

Markov decision processes have been widely applied to medical decision problems including organ transplantation, HIV treatment, cholesterol management and cancer diagnostics. As an illustration we describe an a decision problem that arises when a patient needs a liver transplant [1].

We focus on the decision process for a patient with end-stage liver disease (ESLD)[2] who requires a liver transplant. Organs intermittently become available[3] and vary in quality. Depending upon the quality of the organ, the patient's medical team may either accept the organ and transplant it immediately, or reject it and wait for a higher quality organ.

To make this concrete, a relatively healthy patient may reject a lower-quality organ in the hopes of being offered a higher-quality organ in the future. On the other hand, a patient in poor health may accept the first liver available. A Markov decision process model can be used to formalize this decision problem and explore trade-offs.

To facilitate modeling, patient health status and organ quality are represented by discrete categories. We order health states from 1 to $H$ where 1 represents the healthiest state and $H$ represents the least healthy state. The state $\Delta$ represents death by liver failure. Similarly, liver quality states are ordered from 1 (highest) to $L$ (lowest) with state $\Phi$ representing the case where no liver is offered.

Rewards measure life expectancy in days. On a day when there is no transplant, the patient accrues a reward of 1 (an extra day of life) independent of the health state. The life expectancy post-transplant of a patient in health state $h$ who accepts a liver of quality $l$ is represented by $R(h, l)$. On would expect that that $R(h, l)$ is non-increasing (decreasing) in $h$ and $l$.

We assume decisions are made at the start of each day. When a patient does not receive a transplant, the patient's health state either remains the same or deteriorates. Moreover whether or not a patient is offered a liver depends on a patients health state. For each decision epoch, let

- $q(h'|h)$ denote the probability that a patient in health state $h$ deteriorates to state $h'$ $h' = h, h+1, \ldots, H$,
- let $\delta(h)$ denote the probability a patient in health state $h$ dies from liver failure,

[1] This application is adopted from Alagoz et al. [2007]

[2] *End-stage liver disease* or *cirrhosis* refers to a condition in which a patient's liver is severely damaged and no longer able to function adequately. Without a transplant, it is usually fatal.

[3] From a recently deceased individual. These are referred to as *cadaveric* organs in the medical literature.

- let $w(l|h)$ denote the probability that a patient in health state $h$ is offered a liver of quality $l = 1, \ldots, L$ at the start of a decision epoch.
- $\phi(h)$ denote the probability a patient in health state $h$ is *not* offered a liver in a period.

Assume these distributions are stationary and independent. An Markov decision process formulation follows.

**Decision Epochs:** Assume decisions are made daily and that the horizon is infinite[4] Thus,

$$T = \{1, 2, \ldots\}.$$

**States:** States represent the patient's health (if alive) and liver quality (if available). We add an absorbing state $\Gamma$ to represent the post-transplant state. For convenience, we define $S_H = \{1, \ldots, H\}$, $S_L = \{1, \ldots, L\}$ and $S_L^+ = S_L \cup \{\Phi\}$. Then,

$$S = (S_H \times S_L^+) \cup \{\Delta, \Gamma\}.$$

Note there is no need to distinguish $\Delta$ and $\Gamma$ since the decision process ends in either case. We do so for clarity.

**Actions:** Let $a_t$ represent the action to accept an organ (assuming one is available) and $a_w$ represent the action to wait, (do not accept an organ). We also let $a_w$ represent the "do nothing" action, which applies in the death state, in the post-transplant state, and in any state when no organ is offered. Hence

$$A_s = \begin{cases} \{a_t, a_w\} & s \in S_H \times S_L \\ \{a_w\} & s \in (S_H \times \Phi) \cup \{\Delta, \Gamma\}. \end{cases}$$

Note we must use $a_w$ if no organ is available, the patient has died or a transplant has occurred. We can only use $a_t$ if the patient is alive and an organ is available.

**Rewards:** The reward function may be represented for $s = (h, l)$ as:

[4] A practical upper bound on the number of decision epochs might be 26,000 (assuming no transplants for people over 90 or younger than 20). Given this upper bound, an infinite horizon model may be appropriate and simpler, especially since the process will reach an absorbing state eventually, either post-transplant or death most likely before reaching 90 years old.

$$r(s,a,j) = \begin{cases} R(h,l) & (h,l) \in S_H \times S_L,\ a = a_t,\ j = \Gamma \\ 1 & (h,l) \in S_H \times S_L^+,\ a = a_w,\ j = (h',l') \in \{h,\ldots,H\} \times S_L^+ \\ 0 & (h,l) \in S_H \times S_L^+, a = a_w,\ j = (h',l') \in \{0,\ldots,h-1\} \times S_L^+ \\ 0 & (h,l) \in S_H \times S_L^+, a = a_w,\ j = \Delta \\ 0 & s \in \{\Delta,\Gamma\},\ a = a_w,\ j = s. \end{cases}$$

Note the second and third expressions express the assumption that the health state cannot improve and that the patient does not accrue an additional day of life if death occurs on the current day.

**Transition Probabilities:** If the patient does not receive a transplant, then a transition from health state $h$ to state $h'$ with $h' \geq h$ may occur. If the patient receives a transplant, then the transition is deterministic to the post-transplant state, $\Gamma$. Similarly, if the action is to "do nothing" in the post-transplant or death state, then the transition is a deterministic self-transition.

$$p(j|s,a) = \begin{cases} q(h'|h)w(l|h) & s = (h,l) \in S_H \times S_L^+,\ a = a_w,\ j = (h',l') \in \{h,\ldots,H\} \times S_L \\ q(h'|h)\phi(h) & s = (h,l) \in S_H \times S_L^+,\ a = a_w,\ j = (h',l') \in \{h,\ldots,H\} \times \Phi \\ \delta(h) & s = (h,l) \in S_H \times S_L^+,\ a = a_w,\ j = \Delta \\ 1 & s \in S_H \times S_L, a = a_t, j = \Gamma \\ 1 & s \in \{\Delta,\Gamma\}, a = a_w, j = s \\ 0 & \text{otherwise.} \end{cases}$$

**Application Challenges**

Application of Markov decision process models to clinical decision making requires medical domain knowledge including the nature and progression of the disease, and the treatment options and processes. For instance, applying this model to liver transplantation requires in-depth knowledge of the liver transplant system, the process used to allocate organs and the progression of end-stage liver disease. Data required includes discretized patient health and liver quality states, an estimate of post-transplant life expectancy, probabilities of health state deterioration, and arrival distributions of organs for transplantation by quality. Such data may be obtained from transplant centers and organizations that manage the transplantation system, such as the United Network for Organ Sharing (UNOS) in the United States. Patient health status can be measured using the Model for End Stage Liver Disease (MELD) score, which is a function of various laboratory values. The scores range from 6 to 40, with higher scores indicating poorer health and a higher mortality rate[5]. When data is sparse, MELD scores can be

[5]Internet sources suggest that the 3-month survival rate of 27% in patients with MELD score of 40 and 98% for patients with MELD score between 1 and 9.

aggregated or smoothed. A similar approach can be taken when defining liver quality states, which may depend on donor age, race and sex. UNOS data may be used to estimate $R(h, l)$ (days of survival post-transplant) with a proportional hazards model[6], and organ arrival rates. Transitions between health states can be modeled using a natural history model[7].

## 2.6 Advance Appointment Scheduling

In many applications, decision makers must allocate scarce resources prior to the arrival of future random demand. As an example, a hospital diagnostic imaging department faces the challenge of scheduling appointments for current medical imaging requests so as to meet clinical wait time targets, without knowing exactly how many and when future requests will arrive.

Suppose appointment requests arrive throughout the day and at the end of each day a radiologist assigns each request to one of $K$ *urgency classes*. Urgency class $k$ is associated with a target wait time of $T_k$ days, $k = 1, 2, \ldots, K$, which is chosen based on clinical considerations. A patient in urgency class $k$ should be scheduled prior to day $T_k$. The urgency classes are ordered, with 1 having the highest priority, so $T_1 < T_2 < \cdots < T_K$. If a patient in urgency class $k$ is scheduled after $T_k$, a cost $C_k$ is incurred proportional to the number of days past $T_k$ the appointment is scheduled. This cost can be thought of as a penalty related to worse clinical outcomes due to the delayed imaging. We assume that $C_1 > C_2 > \cdots > C_K$ to represent that the higher delay cost are associated with more urgent cases. If a patient in class $k$ is scheduled before the target $T_k$, no cost is incurred.

Let $p_k(w), k = 1, 2, \ldots, K$ denote the probability that $w$ new class $k$ appointments arrive each day. Let $\mathcal{W} = \{0, 1, \ldots, M\}$ denote the set of possible values of $w$. To ensure a finite formulation, we assume $M$ is finite. Daily capacity is divided into $B$ appointment slots. This means that at most $B$ regular time appointments can be booked each day. We assume there are an unlimited number of overtime slots available and the system incurs a cost of $h$ for each patient scheduled to overtime. Implicit is that $h > C_1$, which means that delaying a patient by a day beyond the target time is less costly than scheduling the patient to overtime. But for a sufficiently large delay, the cost of overtime will be less than the cost of delay.

Once all requests have an assigned urgency class, a scheduling clerk assigns an appointment date to each request and informs the patient. Figure 2.6 provides a timeline for this process. The challenge is to schedule today's requests before realizing future requests for appointments in the face of limited capacity.

[6]This is a statistical model that can be used to determine the impact of covariates on survival times when some patients in the data set are still alive.

[7]A commonly used epidemiological model that simulates disease progression accounting for different risk factors.

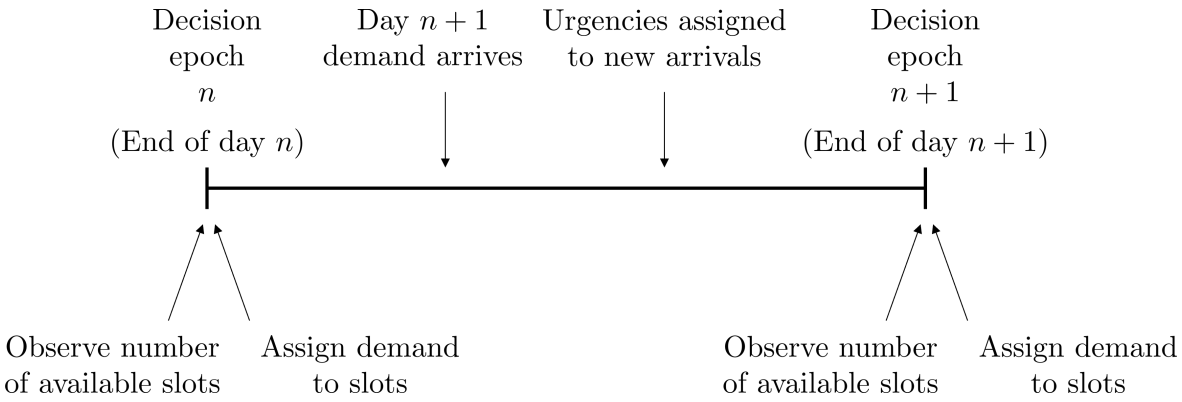


Figure 2.6: Timing of events in the appointment scheduling model.

There are numerous issues arising in this and other similar scheduling problems that can impact modeling:

1. Does the system have access to surge capacity or overtime?
2. How does appointment length vary between patients?
3. Can overbooking be used to account for patient no-shows and late cancellations?
4. Can appointment dates be changed after scheduling?
5. Are the targets flexible or must they be met?
6. Does demand have any seasonal patterns or correlation across urgency classes?
7. How far in advance can appointments be scheduled?

In our example, we assume: 1) access to overtime, 2) fixed appointment lengths, 3) no no-shows / late cancellations 4) no rescheduling, 5) flexible target dates (but with a penalty for exceeding the target date), 6) stationary arrivals and uncorrelated demand between urgency classes, 7) a fixed appointment *booking horizon* of $N$ days. Note that the booking horizon refers to how far into the future current appointment requests can be scheduled, and not the length of the planning horizon. As an alternative to overtime, appointments not scheduled during a particular day may be held over for scheduling in the future at some cost. Exercise 15 considers this variation.

**Decision Epochs:** Decision epochs correspond to the time in the day when the scheduling clerk assigns an appointment date to each appointment request waiting to be scheduled. This is naturally modeled as an infinite horizon problem, so

$$T = \{1, 2, , \ldots\}.$$

**States:** A typical state of the system is represented by $s = (b_1, b_2, \ldots, b_N, w_1, w_2, \ldots, w_K)$. Let $b_i \in \mathcal{B} = \{0, 1, \ldots, B\}$ denote the number of appointments that have already been booked on day $i$ for $i = 1, \ldots, N$. Let $w_k \in \mathcal{W}$ denote the number of appointments of urgency class $k$ waiting to be scheduled at a decision epoch. Note that if there are $b_i$ appointments booked on day $i$, then there are $B - b_i$ remaining appointment slots on that day.

To simplify notation, we introduce the vectors $\mathbf{b} := (b_1, b_2, \ldots, b_N)$ and $\mathbf{w} := (w_1, w_2, \ldots, w_K)$. Thus, a specific state is denoted by $(\mathbf{b}, \mathbf{w})$, and the state space is

$$S = \mathcal{B}^N \times \mathcal{W}^K.$$

**Actions:** Actions represent the number of waiting patients in urgency class $k$ to schedule on each day within the booking window and possibly through overtime if $B$ appointments have already been scheduled. Let $x_{kn}$ denote the number of class $k$ patients to book on day $n$ and $y_k$ denote the number of class $k$ patients to book for overtime the next day. Note that it is not necessary to consider booking overtime slots farther out since we assume access to unlimited overtime, and booking overtime further in the future would simply increase costs.

Define the vectors $\mathbf{x} := (x_{11}, \ldots, x_{1N}, x_{21}, \ldots, x_{2N}, \ldots, x_{K1}, \ldots, x_{KN})$, $\mathbf{y} := (y_1, \ldots, y_K)$ and $\mathbf{0} := (0, 0, \ldots, 0)$ with length $NK$, $K$ and $(N+1)K$, respectively. The action set $A_{(\mathbf{b},\mathbf{w})}$ is

$$A_{(\mathbf{b},\mathbf{w})} = \left\{ (\mathbf{x}, \mathbf{y}) \geq \mathbf{0} \;\middle|\; \sum_{n=1}^{N} x_{kn} + y_k = w_k \text{ for } k = 1, \ldots, K \text{ and } b_n + \sum_{k=1}^{K} x_{kn} \leq B \text{ for } n = 1, \ldots, N \right\}. \quad (2.6)$$

The first condition in (2.6) ensures that all class $k$ requests must be scheduled either to a specific day or to overtime. The second condition ensures that at most $B$ patients may be scheduled to regular time each day.

**Rewards:** We can write the penalty cost associated with scheduling an urgency class $k$ request $n$ days from the current day as $C_k(n - T_k)^+$. This function specifically models the cost being linear in the number of days a class $k$ appointment is scheduled beyond its target, while incurring zero costs for scheduling prior to the target. Exercise 16 considers the variation where instead of the target representing a fixed day, a target window is used.

The reward for choosing actions $(\mathbf{x}, \mathbf{y})$ in state $(\mathbf{b}, \mathbf{w})$ is

$$r((\mathbf{b}, \mathbf{w}), (\mathbf{x}, \mathbf{y})) = -\sum_{k=1}^{K} \sum_{n=1}^{N} C_k(n - T_k)^+ x_{kn} - h \sum_{k=1}^{K} y_k.$$

The reward (negative of cost) captures the costs associated with exceeding the target times as well as overtime costs for the current set of appointment requests. Notice that the reward does not depend on the next state after $(\mathbf{b}, \mathbf{w})$ since the updated $\mathbf{b}$ vector is entirely determined by the current actions and the cost of future appointment requests will be captured in the reward of the new state with an updated $\mathbf{w}$ vector.

**Transition Probabilities:** Transition probabilities depend on both the action choice and the random arrival distribution. At the start of a period, the calendar moves forward one day so previous bookings that were $n$ days from the previous decision epoch are now $n-1$ days from the current decision epoch. Added to these bookings are the newly arriving demand that is booked over the $N$-day horizon starting at the current decision epoch. Finally, since there were no bookings $N+1$ days from the previous decision epoch, there are 0 appointments booked $N$ days from the current decision epoch. The demand that has arrived since the last decision epoch is the only stochastic element in this model. Once this random demand has been assigned to urgency classes, the probability transitions are:

$$p((\mathbf{b}', \mathbf{w}') \mid (\mathbf{b}, \mathbf{w}), (\mathbf{x}, \mathbf{y})) = \begin{cases} \prod_{k=1}^{K} p_k(w'_k) & \mathbf{b}' = (b_2 + \sum_{k=1}^{K} x_{k2}, b_3 + \sum_{k=1}^{K} x_{k3}, \ldots, b_N + \sum_{k=1}^{K} x_{kN}, 0) \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

Some comments about this formulation follow:

- The concept of a booking horizon may be regarded as an artifact of the modeling process and imposed to maintain a finite state space. We can use a fixed booking horizon since our model considers the availability of unlimited overtime. Without overtime, an unbounded booking horizon may be needed.

- Because the booking horizon remains constant between decision epochs, but moves forward each day, the model may be regarded as using a *rolling horizon* model.

- Note that the transitions decompose into a deterministic part corresponding to the number of booked appointments each day and a random part corresponding to the random demand for each urgency class.

- When the maximum daily demand for each urgency class is $M$, the model has $(C+1)^N(M+1)^K$ states. This makes direct computation infeasible for practical sizes of these parameters and motivates the need for approximation (see Chapter **??**).

- After an action is implemented at decision epoch $n$, the $\mathbf{b}$ component of the state does not change until after decision epoch $n+1$. For reasons discussed in Section

**??** and Chapter **??**, it may be more convenient to formulate the model in terms of post-decision states.

- In reality, many possible reward structures may be applicable for this problem. For example, as an alternative to incurring costs for appointments scheduled beyond their target date, a decision maker could strive to maximize the fraction of patients scheduled within their target dates.

### Application Challenges

Application challenges include specifying a booking horizon, specifying how unscheduled cases are dealt with, determining urgency classes and targets, specifying costs for delays, and estimating demand.

In real problem settings, a booking horizon may be determined by the decision maker based on their typical clinical processes. It has also been shown that when unlimited appointment diversion is possible, for example through overtime, an optimal policy is independent of the booking horizon provided it exceeds the largest wait time target. Urgency classes should be defined based on clinical guidelines. The above model formulation was based on a real application, and in that application the classes were "urgent" (7 day target), "semi-urgent" (14 day target) and "non-urgent" (28 day target). Emergency cases were scheduled to a different resource. Relative delay costs can potentially be quantified by calculating the impact on clinical outcomes of delayed treatment due to the delayed imaging.

Future demand may be forecasted using historical data. In practice, these distributions may be non-stationary as volumes generally increase over time. As an alternative to estimating the demand for each urgency class separately, if the historical data suggests that the relative proportion of cases from each urgency class is stable, it may be best to estimate total demand and then split it among each class based on the fixed proportion.

## 2.7 Grid World Navigation

*"A mathematician is a machine for turning coffee into theorems."*
*Alfred Renyi, Mathematician, 1921-1970*

A working mathematician frequently requires coffee and, to avoid time away from theorem proving, employs a robot to bring coffee when needed. The robot's task is to carry the mathematician's empty cup from the office to the coffee room, fill the cup with coffee and bring it back to the mathematician's office, all while avoiding falling down an open stairwell. Figure 2.7 depicts the grid the robot must navigate to retrieve and deliver coffee.

We assume that the robot knows the arrangement of the grid, which grid cell it occupies and the location of the grid boundaries. This means the robot will never

attempt to move outside the grid boundary. Variations of this problem may consider the situation where the robot does not know its location with certainty or the configuration of the grid. Section 3.4.5 describes one such example.

In each cell except the stairwell, regardless of whether the coffee cup is empty or full, the robot can move in any of the four directions that does not take it outside the grid boundary. If the robot falls into the stairwell, or returns to the office with a full coffee cup, it does not move further. We assume movement on the grid is subject to uncertainty as follows. Let $p_E$ and $p_F$ denote the probability that the robot moves in its intended direction when the coffee cup is empty and full, respectively. If in some state there are $k$ possible locations where the robot can end up (including the robot staying in the same location), then the probability the robot moves in each unintended direction or remains where it was is $(1 - p_E)/(k - 1)$ or $(1 - p_F)/(k - 1)$, depending on the status of the coffee cup. For example, suppose the robot has a full coffee cup and is in cell 6. Then if it intends to move down, it does so with probability $p_F$ and it moves left, up or remains in the same location with probability $(1 - p_F)/3$. The robot is more likely to be error prone with a full coffee cup because of the energy required not to spill the coffee. Thus, we assume $p_E > p_F$.

The goal is to return with a full coffee cup at the earliest possible decision epoch. If the robot successfully delivers coffee to the mathematician it receives a reward of $B$ . If it falls down the stairs it incurs a penalty of $X$ because the mathematician has to interrupt work to rescue the robot. We assume $X \gg B \gg 1$.

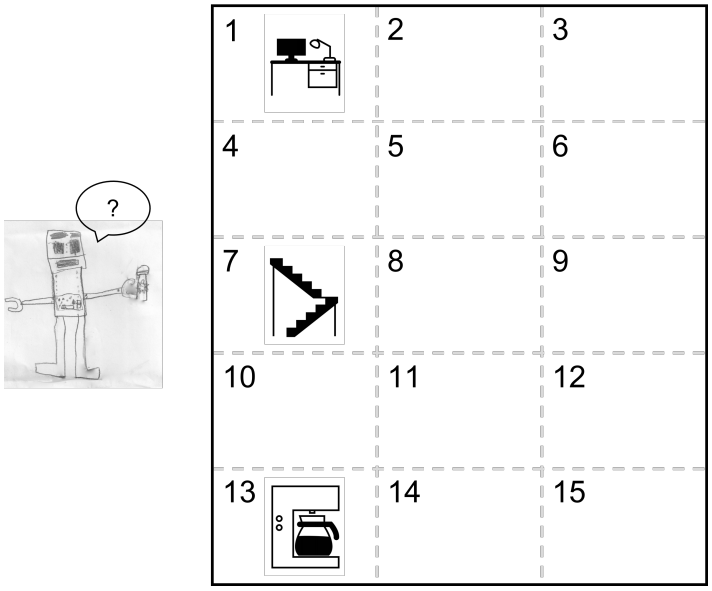


Figure 2.7: Schematic layout for grid world navigation example.

**Decision Epochs:** We assume the process evolves in discrete time, where decision epochs correspond to the instant at which the robot decides in which direction to move. The first decision epoch after the robot enters the coffee room is used to fill up the cup, and the next one corresponds to the instant immediately after the cup has been filled and it decides where to move next.

$$T = \{1, 2, \dots\}.$$

The set of decision epochs is unbounded because the robot continues attempting to deliver the coffee to the mathematician until it is either successful or falls down the stairs.

**States:** States represent the location of the robot in the numbered grid, plus an additional variable to indicate whether or not the coffee cup is empty ($E$) or full ($F$). The status of the coffee cup is required because it informs the success probability of an intended action. It also influences the direction in which the robot should proceed: a robot with an empty coffee cup seeks the coffee room, while a robot with a full coffee cup seeks the office. Therefore,

$$S = \{1, 2, \dots, 15\} \times \{E, F\}.$$

For convenience, we will refer to the office as $O$ (grid cell 1), the stairwell as $ST$ (grid cell 7), and the coffee room as $CR$ (grid cell 13).

**Actions:** Actions represent the direction the robot attempts to travel. Assume the robot can only move north, south, east and west, denoted by $U$, $D$, $R$ and $L$, respectively. Because we assume the robot knows the layout of the grid and its location, the grid boundary constrains its intended movement. For example,

$$A_{(5,\cdot)} = \{U, D, R, L\},\ A_{(7,\cdot)} = \{U, D, R\},\ \text{and}\ A_{(15,\cdot)} = \{U, L\}.$$

We distinguish action sets for some particular states as follows:

$$A_{O,F} = A_{ST,F} = A_{ST,E} = \{a_0\},\ A_{CR,E} = \{a_1\}.$$

The states $(O, F), (ST, F)$ and $(ST, E)$ are *absorbing* states. Once entered, the robot remains there forever; action $a_0$ corresponds to remaining in that state. Once such a state is entered, decision making stops. When the robot enters the coffee room with an empty cup, we assume it takes one period to fill it. We denote the action of filling the cup by $a_1$.

**Rewards:** Each intended movement action and the act of filling the coffee cup costs $c = 1$ time period. Transitions into the office with a full coffee cup result in a reward of $B - c$. For example,

$$r((2, F), a, (O, F)) = B - c, \quad a \in A_{(2,F)} \tag{2.8}$$

Transitions into the stairwell receive a reward of $-X-1$ regardless of the status of the cup. For example,

$$r((4,k),a,(ST,k)) = -X-c, \quad a \in A_{(4,k)} \text{ and } k \in \{E,F\}. \tag{2.9}$$

All other feasible state transitions between neighboring cells result in a reward of $-c$. For example,

$$r((5,k),a,(6,k)) = -c, \quad a \in A_{(5,k)} \text{ and } k \in \{E,F\}. \tag{2.10}$$

Finally, no rewards are received (or costs incurred) once the robot completes its task or falls down the stairs:

$$r((O,F),a_0,(O,F)) = 0 \text{ and } r((ST,k),a_0,(ST,k)) = 0, \ k \in \{E,F\}. \tag{2.11}$$

Note that in the absence of uncertainty, the robot can complete its task in 13 steps, so the maximum possible reward is $B-13$.

**Transition Probabilities:** We provide some typical probabilities:

$$p((k,F)|(5,F),D) = \begin{cases} p_F & k = 8 \\ (1-p_F)/4 & k \in \{2,4,5,6\} \\ 0 & k \notin \{2,4,5,6,8\} \end{cases}$$

$$p((k,E)|(6,E),U) = \begin{cases} p_E & k = 3 \\ (1-p_E)/3 & k \in \{5,6,9\} \\ 0 & k \notin \{3,5,6,9\} \end{cases}$$

Transitions are deterministic when the system is in one of the absorbing states or when the robot enters the coffee room with an empty cup (since the only action is to fill the cup):

$$\begin{aligned} p((CR,F)|(CR,E),a_1) &= p((O,F)|(O,F),a_0) \\ &= p((ST,F)|(ST,F),a_0) = p((ST,E)|(ST,E),a_0) = 1. \end{aligned}$$

Some comments about this example follow:

1. This stylized example is in the spirit of numerous problems that have appeared in the reinforcement learning literature on robotic control. It combines features of stochastic shortest path and gambler's ruin problems.

2. A key feature of this model is that the robot must trade off between safe but slow and risky but fast policies. This type of trade-off is characteristic of the key tension in many applications modeled by Markov decision processes. Here, by trying to reach the coffee room and returning to the office by the shortest route, the robot risks a high probability of falling down the stairs. If robot motion with an empty cup does not involve any randomness, that is $p_E = 1$, the robot will travel from the office to the coffee room by the shortest path but most likely take a more cautious path when returning to the office with a full cup.

**Application Challenges**

The above example is artificial but Markov decision processes have been widely applied to robotic control. Realistic challenges include modeling uncertainty in intended movement, specifying the behavior of the robot's sensors, and providing the robot with a mapping of the area. These challenges are amplified when the application involves robotic movement in a three-dimensional space.

## 2.8 Optimal Stopping

An elegant collection of applications belongs to the class of problems known as "optimal stopping problems". Examples include selling an asset, finding a parking spot, or online dating. We describe these examples after formulating the general model. Optimal stopping problems have attracted considerable research effort, which has primarily focused on showing that an optimal policy has an intuitively appealing structure.

In optimal stopping problems, the system evolves as a (possibly non-stationary) Markov chain on a set of states $S'$ with transition probabilities $b_n(j|s)$ for $s \in S'$ and $j \in S'$ at epoch $n$. If the decision maker decides to "stop" in state $s$ at decision epoch $t$, the decision maker receives a reward of $g_n(s)$. If the decision maker decides to "continue," the decision maker incurs a cost $f_n(s)$. In the finite horizon case, when the problem terminates after $N$ decision epochs, the decision maker receives a reward $h(s)$ if the Markov chain is in state $s$ at epoch $N$.

### 2.8.1 Model Formulation

**Decision Epochs:** As noted above, this can be either a finite or infinite horizon model, so

$$T = \{1, \ldots, N\}, \quad N \leq \infty.$$

**States:** The state space is the union of $S'$ and a state $\Delta$ that denotes the stopped state:

$$S = S' \cup \{\Delta\}.$$

**Actions:** Let the action $C$ (for *continue*) denote the decision to not stop and $Q$ (for *quit*) represent the stopping decision. Then the action set is

$$A_s = \begin{cases} \{C, Q\} & s \in S' \\ \{C\} & s = \Delta. \end{cases}$$

We include the action to continue in the stopped state for completeness.

**Rewards:** The reward does not explicitly depend on the destination state $j$, so for $n < N$

$$r_n(s,a) = \begin{cases} -f_n(s) & s \in S', a = C \\ g_n(s) & s \in S', a = Q \\ 0 & s = \Delta, a = C, \end{cases}$$

with $r_N(s) = h(s)$ for $s \in S$ if the horizon is finite.

**Transition Probabilities:** For $n \leq N$

$$p_n(j|s,a) = \begin{cases} b_n(j|s) & s \in S', a = C, j \in S' \\ 1 & s \in S', a = Q, j = \Delta \text{ or } s = j = \Delta, a = C \\ 0 & \text{otherwise.} \end{cases}$$

### 2.8.2 Optimal Stopping Examples

#### Selling an Asset

A homeowner who is moving to another city has $N$ days to sell a house. Offers arrive throughout the day and by the end of the day, the homeowner has to decide whether to accept the best offer received that day, or wait until the next day for new offers. The set $S'$ represents the set of possible values of the best daily offer for the house, assumed to be around its market value (i.e., bounded) and rounded to the nearest dollar (finite). By waiting until the next day, the homeowner incurs costs $f_n(s)$ related to continued home ownership such as maintenance, mortgage interest, property taxes, and advertising. By accepting the best offer on a given day, the homeowner receives a reward of $s$, minus the costs associated with selling the house, $L_n(s)$, which includes realtor fees and taxes. Hence $g_n(s) = s - L_n(s)$. At day $N$, the homeowner must accept the best offer that day, receiving a terminal reward of $h(s) = s - L_N(s)$. The best offer $j$ at decision epoch $n+1$ is determined by a probability distribution $b_n(j|s)$ that may be conditional on the best offer $s$ in epoch $n$. This distribution may be non-stationary. For example, early in the planning horizon, rejections could signal that the homeowner expects higher offers in the future than the current best offer. Later in the planning horizon, if bidders know the homeowner must sell, the offers might decrease in value.

#### Finding a parking spot

A driver seeks a parking spot as close as possible to a restaurant. Assume the driver can move in one direction only and see only one spot ahead. If the spot is not occupied, the driver may decide to park in it or proceed forward. The probability that any spot is unoccupied equals $p$, independent of all other spots.

The optimal stopping formulation follows. Elements of $S'$ consist of a vector, $(s,k)$, where $s$ indicates the location of the parking spot and $k$ indicates whether the spot is free (1) or not (0). We represent the set of potential parking spots by the set of integers,

with 0 denoting the location of the restaurant, positive numbers denoting locations before the restaurant and negative numbers denoting locations past the restaurant. Thus, $S' = \mathbb{Z} \times \{0, 1\}$. States of the form $(s, 1)$ are the only ones where the action $Q$ corresponding to stopping is available. No rewards are accrued if the individual does not park ($f_n((s, k)) = 0$). When the individual parks, the reward is equal to the distance from the restaurant, so $g_n((s, 1)) = -|s|$. The transition probabilities of the underlying Markov chain are given by

$$b_n((s', k')|(s, k)) = \begin{cases} p & s' = s - 1, k = 1 \\ 1 - p & s' = s - 1, k = 0. \end{cases} \tag{2.12}$$

The above description represents the problem as an infinite horizon problem. It reduces to a finite horizon problem by assuming that:

- The driver will not start looking for a spot until reaching a distance $M$ before the restaurant, and
- Observing that if the driver reaches the restaurant and has not yet found a parking spot, the driver will most certainly take the next free one.

As a result of the second observation, we could consider the state $(0, k)$ as the terminal state, and the epoch when it is reached would correspond to the terminal decision epoch. If at location 0 there is a free spot ($k = 1$), then the driver will park and get the maximal reward of 0. If $k = 0$, the driver will continue. Since the location of the next free spot follows a geometric distribution with parameter $p$, the expected distance from the restaurant to the eventual parking spot will be $1/p$. Thus, the terminal reward is

$$h(0, k) = \begin{cases} 0 & k = 1 \\ -\frac{1}{p} & k = 0. \end{cases}$$

### Online Dating

This example provides a modern take on a classical problem that is often referred to as the *secretary problem*[8]. An individual is searching for dates on a dating app. The dating app shares a brief profile for each potential match, including a photo and description, one at a time. For each potential match, the individual can either "swipe left" to pass or "swipe right" to indicate interest. If the individual swipes left, that profile will not be shown again. If the individual swipes right, a match is made and the two will go on a date. The app offers a free trial until the first match is made or until $N$ profiles have been viewed, whichever comes first. The individual is interested in maximizing the probability of finding the best match during the free trial.

[8]This problem was first formulated by Cayley [1875] in the context of finding an optimal solution for playing a sequential lottery.

Assume that the $N$ potential matches have an unobservable ranking from 1 to $N$, with 1 representing the best match. Through the process of examining the profiles, the individual will be able to rank the candidates seen so far relative to each other in a manner that is consistent with the true ranking. If the best profile is seen, the individual will only know that it is the best seen so far, but will not know whether any future profiles will be better. The order of profiles is completely random, so every permutation of the ranks 1 to $N$ is equally likely.

The following formulation may not be immediately obvious, but is the most succinct way to model the problem. Let $S' = \{0, 1\}$, where the state indicates whether the current profile is the best seen so far (1) or not (0). No rewards or costs are accrued if the individual swipes left ($f_n(s) = 0$). Rewards correspond to the probability of choosing the best candidate and are only received upon stopping. If the individual reaches the last profile, the match is automatically made. The terminal reward is thus $h(1) = 1$ and $h(0) = 0$, since the last profile is either the best profile seen so far or not.

If the individual swipes right on the $n$-th profile, $n \leq N$, and it is not the best profile seen so far, then the probability that the $n$-th profile corresponds to the best match is $g_n(0) = 0$. But if it is the best profile seen so far, then $g_n(1) = n/N$. To see why this is true, let us write out the probability that $g_n(1)$ represents explicitly:

$$\begin{aligned} g_n(1) &:= P(\text{profile } n \text{ is rank 1} \mid \text{profile } n \text{ has the highest rank of first } n \text{ profiles}) \\ &= \frac{P(\text{profile } n \text{ is rank 1} \cap \text{profile } n \text{ has the highest rank of first } n \text{ profiles})}{P(\text{profile } n \text{ has the highest rank of first } n \text{ profiles})} \\ &= \frac{P(\text{profile } n \text{ is rank 1})}{P(\text{profile } n \text{ has the highest rank of first } n \text{ profiles})} \\ &= \frac{1/N}{1/n} = \frac{n}{N}. \end{aligned} \tag{2.13}$$

The second equality follows from the definition of a conditional probability. The third equality is due to the fact that if profile $n$ is the top ranked profile, it must be the top ranked profile within the first $n$ profiles as well. Finally, the fourth equality is due to the fact that the order of the profiles is completely random. So the top ranked profile is equally likely to be in any of the $N$ positions. Similarly, any of the first $n$ profiles is equally likely to be the one with the highest rank.

To determine the transition probabilities, we again appeal to the fact that the order of the profiles is completely random. Thus, regardless of the current state, the probability that profile $n+1$ will be the best among the first $n+1$ is $1/(n+1)$. Thus, the transition probabilities are

$$b_n(j|s) = \begin{cases} \frac{1}{n+1} & j = 1, s \in \{0, 1\} \\ \frac{n}{n+1} & j = 0, s \in \{0, 1\} \end{cases} \tag{2.14}$$

## 2.9 Sports Strategy

Analytical methods have recently found widespread use in sport decision making, providing many opportunities for applying Markov decision processes. Some applications involve decisions made throughout a game while others are situational. Situational decisions such as whether to "go for it" on fourth down in North American football or whether to steal a base or sacrifice in baseball concern a decision in a particular state in a model of the whole game. They reduce to one period problems (Section **??**) when the value function for all games states is estimated from historical data. Other examples such as those described below concern recurrent decisions throughout a game or some portion of it.

### 2.9.1 When to Pull the Goalie in Ice Hockey

In ice hockey, a team has the option of replacing its goalie with an offensive player at any time during a game. This strategy is typically used when a team is behind by one or two goals late in the game in the hopes of tying the score and sending the game to overtime. Doing so can be beneficial since there is a greater probability of scoring when an extra offensive player is in the game. However, pulling the goalie also results in a greater likelihood that the opponent scores, since the goal is undefended. Such a strategy is often employed late in a game in order to achieve a tie and send the game to overtime.

This decision problem is naturally modeled in continuous time but in keeping with the development of the book, we describe a discrete time version. Assume that every $h$ seconds a decision is made whether or not to pull the goalie, and that such a decision is not considered prior to $M$ seconds remaining in the game. Usually, $M$ is on the order of 180.

For concreteness, assume that Team A trails Team B by $g$ goals. Let $p_A$ ($p_B$) denote the probability Team A (Team B) scores one goal in an interval of length $h$ when Team A pulls its goalie and let $w_A$ ($w_B$) denote the probability Team A (Team B) scores a goal in an interval of length $h$ when Team A does not pull its goalie. Naturally, $p_A > w_A$ and $p_B > w_B$. It may also be the case that no team scores during an interval of length $h$, so $p_A + p_B < 1$ and $w_A + w_B < 1$. We assume that $h$ is sufficiently small so the likelihood of scoring more than one goal in that interval is negligible.

**Decision Epochs:** Because decisions are made every $h$ seconds up to $M$ seconds,

$$T = \{1, 2, \ldots, N\},$$

where $N = M/h$, the first decision epoch corresponds to $M$ seconds left in the game, the second corresponds to $M - h$ seconds left in the game, and so on.

**States:** We assume the state keeps track of the goal differential, defined as Team B's goals minus Team A's goals. Let $G$ be the maximum goal differential at which the coach would consider pulling the goalie. Then

$$S = \{0, 1, \ldots, G+1\}.$$

We include the additional state $G+1$ to ensure a finite state space. In our formulation, we regard this state as an absorbing state. If the goal differential reaches $G+1$ the coach of Team A will not consider pulling the goalie anymore. If the score differential returns to $G$ because Team A scored a goal without its goalie pulled, the decision problem starts anew. In practice, $G = 3$. A team would not pull its goalie if it is leading, so negative values are omitted from the state space. The state 0 corresponds to a tie score, which also is an absorbing state and the objective of pulling the goalie.

**Actions:** In all states the coach has the option to not pull the goal (action $a_0$). We only consider pulling the goalie (action $a_1$) when the goal differential is between 1 and $G$.

$$A_s = \begin{cases} \{a_0\} & s = 0 \text{ or } G+1 \\ \{a_0, a_1\} & s = 1, \ldots, G. \end{cases}$$

**Rewards:** Since the objective is to tie or win by the end of the planning horizon, rewards are only received at termination. So $r_n(s,a) = 0$ for $n < N$ and all $s$ and $a$. The terminal reward is given by

$$r_N(s) = \begin{cases} 0 & s > 0 \\ 1 & s = 0. \end{cases}$$

**Transition Probabilities:** The transitions probabilities are

$$p_n(j|s,a) = \begin{cases} p_A & j = s-1, s = 1, \ldots, G, a = a_0, \\ p_B & j = s+1, s = 1, \ldots, G, a = a_0, \\ 1 - p_A - p_B & j = s, s = 1, \ldots, G, a = a_0 \\ w_A & j = s-1, s = 1, \ldots, G, a = a_1, \\ w_B & j = s+1, s = 1, \ldots, G, a = a_1, \\ 1 - w_A - w_B & j = s, s = 1, \ldots, G, a = a_1, \\ 1 & j = s = 0, G+1, a = a_0, \\ 0 & \text{otherwise.} \end{cases}$$

### Application Challenges

The key parameters in this model are the relative scoring probabilities. They may vary by team and also depend on the whether the opposing team has been assessed a

penalty so that pulling the goalie results in a "two-man advantage" and an increased scoring probability.

The following data comes from the (North American) National Hockey League for the 2013-2020 seasons. When both teams are at full strength (no player is in the penalty box), teams score goals at the rate of 2.25 goals per 60 minutes. When a team pulls its goalie, its scoring rate increases to 6.39 goals per 60 minutes. However, the opposing team's scoring rate increases to 19.16 goals per 60 minutes. Assuming $h = 5$ seconds, these goal scoring rates correspond to $p_A = p_B = 0.003125$, $w_A = 0.008875$ and $w_B = 0.02661$. On average, teams pull their goalie around 4 minutes with a three goal deficit, 2.3 minutes with a two goal deficit, and 1.4 minutes with a one goal deficit. The success rate for pulling the goalie with a one goal deficit is about 14%.

Penalties play a large role in ice hockey. Pulling the goalie when Team B is penalized significantly affects the scoring probabilities, increasing $p_A$ and decreasing $p_B$ relative to the previous values. The same data shows that pulling the goalie when the opposing team has one player in the penalty box increases Team A's scoring rate to approximately 12 goals per 60 minutes, and decreases Team B's scoring rate to approximately 11 goals per 60 minutes. Note that this latter quantity is still much higher than Team B's goal scoring rate when Team A doesn't pull its goalie, which is 6.54 goals per 60 minutes.

### 2.9.2 A Handicap System for Tennis

Consider a tennis match between two players of unequal skill level. In order to have a fair (and enjoyable) match, the stronger player (Player B) offers the weaker player (Player A) a handicap. The handicap takes the form of a budget of "credits" that Player A can use to win a point without playing it. To our knowledge, such a system has been yet to be widely applied but conceptually presents an interesting strategic challenge: when should Player A use these credits?

Before describing how such a handicapping system may be employed, we briefly review relevant aspects of tennis scoring. A tennis match consists of games and sets. A player wins a game by scoring four points first, provided that the player "wins by at least two points." That is, if both players have scored three points, the winning score needs to be at least 5-3. A player wins a set by being to first to win 6 games, again with a "win by two game" rule in effect. If the set score reaches 6-6, then a tiebreaker is played, with the winner winning the set by a score of 7-6. Finally, a match is typically the best two out of three sets or best three out of five sets.

A key feature of this scoring system is that it is "hierarchical," the match score decomposes into sets and games, each with its own scoring system. Thus the score with Player A serving may be 1-1 in sets, 5-3 in games and 3-1 (commonly referred to as 40-15) in the current game. Faced with this situation, Player A could use a handicap point to win the game, and hence the set and the match.

Unlike sports such as soccer, in which every goal contributes equally to the final score, not every point is equally valuable in tennis. In fact, a player could win more than

50% of the total points in the match, but still lose the match, due to the hierarchical nature of the scoring system.

We now formalize the problem. If Player A uses a credit at the start of a point, then Player A wins the point, the handicap budget is decremented by one, and the players proceed to play the next point. If Player A decides not to use a credit, then the point is played as usual. Let $p_1$ ($p_0$) be the probability that Player A wins the point on serve (return) if it is played out. We assume the budget is for the entire match and that Player A can use a credit regardless of which player is serving.

**Decision Epochs:** The start of each point is a decision epoch. Given the scoring system described above, the horizon is infinite but with a random stopping time corresponding to the end of the match. Thus

$$T = \{1, 2, \ldots\}.$$

**States:** The state comprises the current match score, $q$, the budget of credits remaining, $b$, and an indicator for the serving player, $k$. Suppose the set of possible match scores is $Q = \{q_1, q_2, \ldots\}$, the starting number of credits is $B$, and $k = 1$ (0) indicates that Player A is serving (Player B is serving). We include two absorbing states, $W$ and $L$, that correspond to Player A winning and losing the match, respectively. Thus, the state space is

$$S = (Q \times \{0, 1, \ldots, B\} \times \{0, 1\}) \cup \{W\} \cup \{L\}. \tag{2.15}$$

Each state $q$ represents a set score, a game score and a within-game score. Note that $B$ is upper bounded by 24 times the number of sets to win the match. In a best two out of three sets match, $B$ is bounded by 48 and in a best three out of five sets match $B$ is bounded by 72.

**Actions:** Actions are to use a credit ($a_1$) or not ($a_0$) at each decision epoch when there are credits remaining. Once no credits remain, the only action is $a_0$.

$$A_s = \begin{cases} \{a_0, a_1\}, & s = (q, b, k) \text{ with } b > 0 \\ \{a_0\}, & \text{otherwise.} \end{cases} \tag{2.16}$$

**Rewards:** Since Player A's objective is allocate handicap credits so as to maximize the probability of winning the match, the only non-zero reward is when there is a transition to state $W$, in which case the reward equals 1. Thus

$$r(s, a, s') = \begin{cases} 1, & \text{if } s \neq W, a \in A_s, s' = W \\ 0, & \text{otherwise.} \end{cases} \tag{2.17}$$

Similar to the online dating application, to maximize the probability of an event, the model formulation should be set up so that decision maker receives a reward of 1 only when that event occurs. This follows directly from the observation that the expected value of an indicator variable equals the probability of the indicated event.

**Transition Probabilities:** If a credit is used, then there is a deterministic transition to the state in which Player A has one extra point. Using such a credit could also affect the game and set score, and even result in winning the match. Otherwise, the transitions follow the point-winning distribution of Player A against Player B. For convenience, let $q^+$ ($q^-$) denote the score if Player A wins (loses) the point when the current score is $q$. Similarly, let $k_q^+$ ($k_q^-$) denote the server if Player A wins (loses) the point when the current score is $q$ and the current server is indicated by $k$. The server changes only if by using the credit, Player A wins the current game. Thus, given that the current state is $s = (q, b, k)$,

$$p(j|s,a) = \begin{cases} 1, & \text{if } j = (q^+, b-1, k_q^+), a = a_1 \\ p_k, & \text{if } j = (q^+, b, k_q^+), a = a_0 \\ 1-p_k, & \text{if } j = (q^-, b, k_q^-), a = a_0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.18}$$

### Application Challenges

The primary challenge in the application of this model revolves around the estimation of the point-win probabilities $p_0$ and $p_1$. Such probabilities depend on several factors include the strength of the opponent (perhaps considering the specific opponent and previous head-to-head successes), the court surface, the weather, and recent playing history. These probabilities are likely to be non-stationary as well due to factors like fatigue or injury.

## 2.10 The Art of Modeling

One learns to formulate Markov decision processes by studying how others have done so and practicing themselves. Formulations that appear particularly crisp are likely the result of numerous iterations in formulating and re-formulating the problem. By being exposed to and working through many different examples, one starts to build an intuition for how certain problem types are formulated.

### How to formulate a Markov decision process model

- *Clearly define the problem.* Verbally describe what the decision maker wishes to achieve, what information is available on which to base decisions, how the system responds to these decisions, and what rewards or costs are incurred as

a consequence of the decisions taken. A precise problem description facilitates identifying all model components. Often, however, one must return to, revise or redefine problem characteristics to ensure that the Markov decision process model properly represents the specified situation. In our examples below, we present problem statements that are self-contained, with the information needed to fully formulate the problem.

- *Draw a timeline of events.* Carefully determine *when* the **state** information becomes available, **actions** are chosen (i.e., the **decision epochs**), **rewards** are received and **transitions** occur. Changes to the timing of events can impact the specification of certain model components. Several examples in this chapter illustrate the sequence of events in a typical period. Selected problems at the end of the chapter ask you to reformulate models under modified assumptions about the timing of events.

- *Identify decision epochs.* Specify the precise time at which actions are chosen, using the timeline as a guide.

- *Determine the planning horizon.* Applications may have a horizon that is finite with fixed length, finite with variable length or infinite. Variable length models arise when the policy or realization of a probability take the system to an absorbing state such as in the the lion hunting model (Section 2.4), liver transplant model (Section 2.5), the Grid World model (Section 2.7) and the optimal stopping models (Section 2.8). Most infinite horizon models may be transformed into finite horizon problems through an appropriate reformulation or simply through truncation. An example is provided in the parking problem (Section 2.8.2).

- *Identify states.* The main challenge when formulating a Markov decision process is determining an appropriate state space. Doing so requires taking into account all other model components. Hence, this is the most important step. A well-defined state space can make the rest of the formulation appear obvious; a poorly defined state space may result in complicated (non-Markovian) dynamics, extra computational burden, or insufficient information to write down a complete model. The advance appointment scheduling (Section 2.6) and online dating application (Section 2.8.2) illustrate two challenging examples of state space formulation.

  States should encapsulate all of the information available to the decision maker (and no more than is necessary) to specify actions, rewards and transition probabilities. Be sure not to include actions as part of the state space. Often, a time component is required for decision making, but the decision epoch itself may be sufficient to avoid including an extra variable in the state space. Some models may require the addition of zero-reward absorbing states to account for early or random termination times.

- *Specify actions.* Be sure to note whether different sets of actions may be available in each state. Since the Markov decision process formulation requires specifying an action for each state, in states where there is no meaningful action choice such as an absorbing state, the set of actions should be specified as a single element representing the "do nothing" action.

- *Determine rewards.* Rewards may be stationary or vary with decision epoch. In finite horizon models, be sure to specify a terminal reward function. Also, note whether rewards depend on the subsequent state. It may be possible to model a problem both ways, with rewards that do or do not depend on the next state. However, usually one of these two is more natural. When the reward does not depend on the subsequent state, we will leave it out of the notation.

  In cases when the decision maker seeks to maximize the probability of an outcome, such as surviving or winning, specify a reward of zero in all states not corresponding to that outcome and a reward of one when that outcome occurs. The reason for this is that the expected value of an indicator of an event equals the probability of the event occurring.

  While a Markov decision process is generally concerned with maximizing rewards, its formulation, and the reward function specifically, should be independent of the specific choice of optimality criterion such as expected total reward, expected discounted total reward, long-run average reward, or expected utility. Also, note that in many applications, a decision maker may seek to minimize costs, which can be regarded as negative rewards. Since our Markov decision process formulation seeks to maximize rewards, we regard costs as negative rewards.

- *Specify transition probabilities.* These are often quite complicated and contain many special cases. It is important to appeal to the timeline and the order of events when writing down the transitions. Challenges include taking into account "edge cases" at state space boundaries and noting that some components of the state may evolve deterministically, while others may evolve stochastically. In the presence of absorbing states, be sure to note that under the "do nothing" action the system remains in that state with probability one. For completeness, be sure to note zero probability transitions corresponding to impossible combinations of states and actions, which can be captured under the catch-all heading "otherwise".

  Recall that a Markov decision process with a fixed policy results in a Markov reward process that evolves over a Markov chain. Drawing a Markov chain with directed arcs indicating transitions and denoting probabilities on arcs is a simple but effective method to help ensure that all transitions are accounted for (e.g., probabilities leaving a state for a given action sum to 1) and that they make sense (e.g., transitions occur between states as described in the problem statement). With complicated multi-dimensional state spaces, drawing such a picture is often a must. Figure 2.5 provides an example.

- *Estimate model parameters.* Most of the examples in this chapter are abstracted from real problem situations. To apply the models in concrete settings, one must estimate the model parameters such as transition probabilities and rewards.

  In some cases, such as inventory control (Section 2.2), revenue management (Section 2.1 and queuing control (Section 2.3) the transition probabilities may be derived from parametric distributions with the parameters estimated from historical data. When there are no parametric forms for transition probabilities, care must be taken in estimating probabilities because some may be non-zero but very small. In applications such as the lion hunting model (Section 2.4), clinical decision making (Section 2.5) and sports strategy (Section 2.9) one may appeal to the data and literature from those fields to obtain parameter estimates.

  Another challenge when applying models in practice is specifying rewards. In applications where rewards refer to concrete monetary values, specifying rewards can be relatively straightforward. In examples such as lion hunting (Section 2.4), optimal parking and online dating (Section 2.8.2), pulling the goalie (Section 2.9.1), and tennis handicapping (Section 2.9.2) the reward is implicit in the chosen model objective. In other cases, rewards may be derived in consultation with the decision maker.

## 2.11 Bibliographic Remarks

Inventory models (Section 2.2) date back to at least Arrow et al. [1951] and Dvoretsky et al. [1952]. The book of Arrow et al. [1958] is an important early reference and Porteus [2002] provides an overview in book form. Much current research in inventory theory is subsumed under the heading "supply chain management". Section 8.9.2 of Puterman [1994] provides an inventory example of a constrained MDP with a service level constraint.

The model in Section 2.1 is in the spirit of Gallego and van Ryzin [1994] who provide one of the first examples of dynamic pricing. The book of Talluri and van Ryzin [2004] provides a comprehensive overview of revenue management. Dynamic pricing combined with overbooking has been applied extensively in the airline industry where it is referred to as yield management [Smith et al., 1992].

The queuing control models in Section 2.3 have mostly been studied in continuous time. Early references include Yadin and Naor [1967], Heyman [1968], Naor [1969] and Sobel [1969]. Our formulation of the discrete time service rate control model follows de Farias and van Roy [2003] where they use it to illustrate approximate dynamic programming methods.

The lion hunting example in Section 2.4 is adopted from Clark [1987]. That paper provides many of the parameter values described in the Application Challenges section, including the energy storage capacity, daily energy depletion, biomass yield and catch probabilities of gazelles and zebra. The estimate of energy expenditure while hunting

was based on Hubel and et al. [2016]. Edible biomass of other prey listed were taken from Smuts [1979]. Other applications in ecology include Mangel and Clark [1986], Kelly and Kennedy [1993] and Sirot and Bernstein [1996].

Numerous applications of using Markov decision processes in clinical decision making have appeared in the literature. The model we described is based on Alagoz et al. [2007], who focus on liver transplantation. The discussion around application challenges is based on methods they employed to specify their model and estimate parameters. Other examples of clinical decision making using Markov decision processes include Shechter et al. [2008], who consider HIV therapy, and Kurt et al. [2011], who model statin treatments for diabetes patients.

Our formulation of the advance scheduling model in Section 2.6 follows Patrick et al. [2008]. The result about an optimal policy being independent of the booking window if the window is longer than the largest wait time target and if the system has access to unlimited appointment diversion is given in that paper. Sauré et al. [2012] and Goçgun and Puterman [2014] analyze variants of this model. An example of using the impact of delayed treatment to quantify the cost of delayed imaging appointments is given in Sauré et al. [2012] in the context of radiation therapy.

A grid-world model appears in Sutton and Barto [2018]. Such models have been widely used in the computer science community to illustrate Markov decision process and reinforcement learning concepts.

Optimal stopping problems originate with early work of Wald [1947], Wald and Wolfowitz [1948] and Arrow et al. [1949]. Karlin [1962] proposes and solves the asset selling problem. The optimal parking problem appears in Chow et al. [1971]. The online dating (i.e., secretary) problem was first proposed by Cayley [1875] in the context of evaluating a lottery.

Sports applications have appeared broadly. The path-breaking monograph of Howard [1960] introduces many of the key Markov decision process concepts, and contains an example of using Markov decision processes in baseball strategy. Carter and Machol [1971] and Chan et al. [2021] develop value functions in football. The pulling the goalie model in Section 2.9.1 originates with Morrison [1976]. A dynamic programming formulation was provided in Washburn [1991]. Hall [2020] provides the recent data quoted in the Applications Challenge section. The tennis handicapping model in Section 2.9.2 appears in Chan and Singal [2016]. The amazing book by Kemeny and Snell [1960] includes a Markov Chain model for a tennis game.

## 2.12 Exercises

1. Formulate a periodic inventory control problem in which orders arrive after demand has been fulfilled. Clearly show how the timing of events in Figure 2.1 changes.

2. Formulate a periodic inventory control problem in which sales are lost if demand

exceeds supply in a period.

3. Formulate a periodic inventory control model in which there is limited storage capacity, limited backlogging and a bound on order size. Assume if the quantity backlogged exceeds its bound, there is an extra cost incurred.

4. Formulate a service rate control queuing model with a fixed cost $C$ for changing the service rate. Note it is not sufficient to just modify the reward function.

5. Formulate a combined admission and service rate queuing control problem as a Markov decision process.

6. Consider a call center with monolingual (English-only) and bilingual (English and French) call takers. Assume there are two call takers of each type. English-speaking and French-speaking customers call in to the center and indicate their preferred language. English-speaking customers can be served by either type of call taker, but French-speaking customers can only be served by a bilingual call taker. Every minute an English-speaking customer calls in with probability $p_E$ and a French-speaking customer calls in with probability $p_F$, where $p_E + p_F < 1$. We assume the probability of more than one caller per epoch is negligible. Customers incur a cost of $C$ for every minute spent waiting before service. The duration of a call is geometric with parameter $q$, regardless of the language. Formulate this routing control problem as a Markov decision process.

7. (Control of a tandem queuing network) Consider a discrete-time queuing system composed of two single-server queues in tandem and two types of jobs. Type 1 jobs arrive at queue 1 and after completing service at queue 1 also require service at queue 2. Type 2 jobs arrive directly at queue 2 and require service at queue 2 only. **(June 22 insert picture)** In each period, no job arrives or either a type 1 job arrives, a type 2 job arrives or one of each type arrives. Assume the probability of an arrival of a type $i$ job is $p_i$ independent of the whether or not the other type job arrives.

   Assume a finite buffer (waiting room) of size $M_i$ in front of queue $i$. When the buffer is full jobs are blocked (lost) at penalty cost $c_i$. Completion of a type $i$ job yields revenue $R_i$ with $R_1 > R_2$. In addition, assume a holding cost of $h(s_1, s_2)$ when there are $s_i$ type $i$ jobs in the system (either in the queue or in service).

   Formulate the following revenue maximization problems as Markov decision process.

   (a) (Job selection) In each period the controller of queue 2 can choose whether to serve a type 1 or type 2 job. Assume that the service is completed with probability $q_2$ independent of job type and whether or not a job at queue 1 has completed service with probability $q_1$. To simplify the formulation assume that if the service is not completed in the current period, the job reverts to the queue prior to the start of the subsequent period.

(b) (Service rate control) In each period the system can choose the service probability at queue 1 from the set $\{q_{1,1}, q_{1,2}, \ldots, q_{1,N_1}\}$ with cost $f_1(q)$ that is non-decreasing in $q$.

(c) Describe and formulate other possible control problems that can apply to this configuration.

8. Formulate a version of the Grid World Navigation model in which there is a positive probability that the robot drops the coffee cup, which increases if the cup is full.

   (a) Suppose the cup is breakable and if it is dropped, the robot needs to return to the office to retrieve another one.

   (b) Suppose instead that the cup is not breakable and the robot spends one epoch picking up the now empty cup.

   Clearly state any assumptions you are making in formulating this model.

9. Formulate the following single machine maintenance problem. You own a piece of equipment that deteriorates over time. While it is operating, it contributes revenue of $r$ dollars per month. When it has been operating for $i$ months since maintenance, the probability it fails in the current month is $p(i)$ and the probability it does not fail is $1 - p(i)$. If it fails at any time during a month the cost of repairing it is $c_A$ and it is available for use at the start of the subsequent month. Assume that if it fails during a month, no revenue is generated during that month. On the other hand, the maintenance manager can schedule preventive maintenance in a month at cost $c_B$. Assume preventive maintenance is always scheduled at the beginning of the month, starts in the first week of the month and takes one month.

   (a) Draw a timeline for the decision problem and clearly state any assumptions you are making.

   (b) Formulate the maintenance manager's problem as a Markov decision process. Be clear to state any assumptions you make.

   (c) In a real application, how do you think $p(i)$ will vary with $i$ and what would be the relationship between $c_A$ and $c_B$?

   (d) Propose a "real life" application of this model.

10. [Stengos and Thomas, 1980] Consider the following generalization of the previous problem. You own two pieces of equipment which sometime require maintenance that takes three weeks. Maintenance on one machine costs $c_1$ per week while maintenance on two machines costs $c_2$ per week. The probability either piece of equipment breaks down if it has operating for $i$ weeks is $p(i)$. Assume that if the equipment breaks down during a week, maintenance begins at the start of the

next week. However, if you decide to perform preventive maintenance, you do so at the start of a week. Moreover, assume that the two pieces of equipment break down independently.

(a) Formulate this problem as an infinite horizon Markov decision process.

(b) How are $c_1$ and $c_2$ related? Why?

11. Reformulate Exercise 10 assuming that the time it takes to complete maintenance on a piece on equipment is random. In particular, assume that maintenance is completed in any period with probability $q$, independent of other periods.

12. [Bertsimas and Shioda, 2003] A restaurant contains both two-seat and four-seat tables. Parties of two and four arrive randomly and request service. No reservations are taken. When a party of four arrives and a four-seat table is empty, they should be seated but should the manager ever seat a party of two at a four seat table? If so, when? Also, when should requests for service be denied, if ever?

    Formulate this problem as Markov decision process assuming the following, unrealistic as it may be. Decisions are made every 10 minutes and the restaurant operates 24 hours a day. There are two two-seat and two four-seat tables. Meals consist of two courses, durations of each are geometrically distributed independent of party size. Course 1's completion probability per epoch is 0.7, while course 2's is 0.8. In any period there is at most one arriving party. A party of two arrives with probability 0.2 and a party of four with probability 0.1. Assume that the waiting area holds at most 6 people. If it is full, arrivals are blocked and do not enter. Also any waiting party may leave in a 10 minute period with probability 0.05.

    Revenue is as follows. A party of 2 contributes \$50 and a party of 4 contributes \$100. The cost of waiting (incurred by the restaurant) is \$6 per person per hour.

13. Formulate a finite horizon Markov decision process as an infinite horizon model by augmenting the state with the decision epoch and adding absorbing states.

14. In the organ transplantation problem, consider an extension where outcomes depend on the quality of the organ that is offered. Modify the formulation to account for this possibility.

15. Formulate the advance scheduling problem when appointments not booked on first day available are added to the next days demand with cost $C'$.

16. Formulate the advance scheduling problem where instead of the target representing a fixed day, target windows are used for each urgency class. Let $T_k^l$ and $T_k^u$ be the lower and upper limits of the target window for urgency class $k$. If an appointment is scheduled within this window, no costs are incurred. If a class $k$ appointment is scheduled before (after) $T_k^l$ ($T_k^u$), then a cost of $C_k^l$ ($C_k^u$) is incurred.

17. Formulate the advance scheduling problem where rewards are accrued if patients are scheduled before their fixed target date.

    (a) Let $d_k$ be the reward for each patient of class $k$ who is scheduled before the target date $T_k$.

    (b) What is the interpretation of the objective if $d_k = 1$ for all urgency classes $k$?

18. Formulate a finite horizon version of the Grid World problem in which if the robot does not return with coffee after $N$ decision epochs, the mathematician gets his or her own coffee and and incurs a penalty of $C$ units. How should $C$ be related to $X$ and $R$?

19. Modify the lion hunting problem to take into account that on any day, the lion may be captured by poachers with probability $1 - \lambda$. How is the related to discounting?

20. At the start of each day, a lion decides whether or not to hunt and if so, in what group size. The probability of catching prey varies with group size. This presents a trade-off, a larger group has a greater probability of a successful hunt, but then less food is available for each lion in the group. Assume a maximum group size of $M$ and that all captured prey is split evenly among the group. Let $\lambda_m$, $m = 1, 2, \ldots, M$, denote the probability that a group of of size $m$ is successful in its hunt. We assume the prey being hunted yields a total edible biomass of $e$ units. Thus if the lion hunts in a group of size $m$ and is successful, it receives $e/m$ units of edible biomass.

21. Reformulate the original lion hunting behavior model so that the objective is to maximize the number of days of survival instead of the probability of survival. Clearly note what changes are necessary.

22. The ride-sharing driver's dilemma. At random times throughout the day, a ride-sharing driver receives offers of potential trips, including their expected revenue and time to complete the trip. The driver can either accept the trip or decline it and wait for the next offer. Formulate this problem as a discrete time Markov decision process clearly stating all assumptions being made.

23. Consider a variant of the online dating problem in which the decision maker's goal is to maximize the probability of choosing one of the two best candidates. How would you modify the formulation to take this into account.

24. Consider a variant of the online dating problem in which the decision maker's goal is to maximize the rank of the selected date. Modify the formulation accordingly.

25. Reformulate the tennis handicapping problem to account for second serves. That is, if a player misses a first serve, they have a second chance to get the ball in before losing the point. Suppose the probability that the first serve goes in is $q_1$ and the probability the second serve goes in is $q_2$. Conditioned on the serve going in, the probability of winning the point is $p_{1,1}$ and $p_{1,2}$ for the first and second serve, respectively. Since first serves tend to be more aggressive, $q_1 \leq q_2$ and $p_{1,1} \geq p_{1,2}$. Assume handicap credits can only be used when serving.

26. "Scrabble, like life, is a trade-off between today and tomorrow-between sending and saving. It's what an economist would call a dynamic programming problem." (Seven Games: A Human History" Oliver Roeder, W.w. Norton Company New York 2022) The game of Scrabble provides an opportunity for applying Markov decision processes. Provide a model for a decision of whether a player should replace some or all tiles during a turn. This is a rather complex model that will be challenging to formulate in its entirety, which we do not believe has appeared in the literature.

27. Formulate the reward function of the inventory management example when the revenue of the inventory sold is included.

**(re-order the questions, but wait until solutions for chap 3 are done)**

# Chapter 3

# Partially Observable Markov Decision Processes

*"Confucious said: To know what you know and what you do not know, that is true knowledge."*

*Henry David Thoreau - from his book Walden, 1854*

In this chapter, we study a class of widely applicable models known as *partially observable Markov decision processes* or *POMDPs*. In a POMDP, the decision maker cannot observe the state of the system but instead receives a *signal* that provides noisy information about the *hidden* system state. Since the state is unobservable, actions must be chosen on the basis of the only available information, namely the sequence of previously observed signals and actions. Unfortunately, this approach quickly becomes unwieldy.

The key idea is this: after observing a signal, Bayes' Theorem can be used to update a prior estimate (probability distribution) of the hidden state to a posterior estimate of the hidden state. Moreover this information is sufficient for decision making. We refer to these probability distributions as *belief states*. Consequently, the state space used for decision making becomes the set of all probability distributions on $S$ which is equivalent to an $|S|$-dimensional unit simplex.

## 3.1 Model overview

Consider a Markov decision process where the decision maker cannot directly observe the system state, but instead can observe a *signal* or *observation* from the system, which provides some information about the true system state. Since action choice can only depend on information the decision maker has, it must depend on previously observed signals and actions, and not the hidden system state.

This type of model is applicable to a wide range of problems. In medicine, outcomes of diagnostic tests serve as signals of the unobservable health state of a patient.

Information from these tests can then inform potential treatment decisions. A similar problem faces the owner of a machine whose true working condition is unknown, but where signals may be observed through sensors or by determining the number of defects in a sample of the machine's output. The owner can use this information to decide whether to perform preventive maintenance. Another example is a robot navigating a foreign environment, where the operator receives a noisy signal of the robot's true position and must make navigational decisions.

Including an observation process allows for two conceptually different models of how information is generated:

1. *Passive information-acquisition models* in which i the action impacts the hidden system so that observations passively provide imprecise information about the system state.

2. *Active information-acquisition models* in which the actions provide information about the system state with varying degrees of accuracy at different costs. In some cases the system state may change and in others it may remain the same.

This distinction will not impact analysis, but provides a conceptual framework for modeling.

Controlling a robot that can move in any one of four directions through a noisy information channel provides an example of a passive information-acquisition model. In it, after each state transition the decision maker receives a noisy signal of the robot's new location and based on it and previous signals and actions decides on the next action. An example of an information-seeking model arises in a clinical setting where the action specifies which of several tests to use to assess a patient's health status. In this case the test does not alter the system state, it just provides information about it. One test may be inexpensive and not very accurate while another test may be invasive and costly but more accurate. If the patient is thought to be in a "good" health state, then the less accurate test may be preferable while if the patient is thought to be in a "poor" health state, the more accurate test may be preferable. Of course, a model can combine passive and information seeking features[1].

Although the hidden states are unobservable, in several applications a subset of states may be observable. Examples include search models in Section 3.4.2 and the cancer screening model in Section 3.4.4. In these models, the observable states correspond to terminal or absorbing states. In the search model, the process terminates when the object is found and in the cancer screening model, the process terminates when treatment starts or the patient dies.

### 3.1.1 Dynamics of a POMDP

We assume the signal belongs to a discrete *signal space* $O$, which in general may depend on the previous history of visited system states and past actions. We focus

[1]See the example in Section 3.4.1 below.

on a simpler case, where the signal only depends on the current system state and the preceding action. Let the random variable, $Z_n$, denote the signal received at decision epoch $n$. We represent its probability distribution by[2]

$$u(o|s,a) := P(Z_{n+1} = o \,|X_{n+1} = s, Y_n = a) \tag{3.1}$$

for $o \in O$, $s \in S$ , $a \in A_s$ and $n \geq 1$. Note that the signal distribution incorporates the action chosen at the *preceding* decision epoch and the state at the *current* decision epoch. We also assume $u$ is stationary, so it does not vary with decision epoch. In passive-information acquisition models, $u$ will be independent of $a$. Note that an MDP is a special case of a POMDP in which there is a one-to-one mapping between observations and states, where $u(o|s,a) = 1$ when $o$ corresponds to $s$ and 0 otherwise.

A POMDP evolves as follows. In decision epoch $n$, the system is in an unobservable state $s^n$ and the decision maker observes a signal, $o^n$. Then, the decision maker chooses an action $a^n$, which leads to an (unobserved) system transition to state $s^{n+1}$ and a reward $r_n(s^n, a^n, s^{n+1})$ (or $r_n(s^n, a^n)$) being accrued. We assume that the decision maker observes the past sequence of actions and signals, but not the sequence of state transitions or rewards since the system state is not observable by the decision maker[3]. Once the system reaches the next state $s^{n+1}$, a new observation $o^{n+1}$ is generated and the process continues. Figure 3.1 illustrates the timing of events in one period.

We assume the decision maker has a prior assessment of the starting state of the process, which is encoded by the probability distribution $b_1(s) := P(X_1 = s)$. By convention, there is no observation at decision epoch 1, and only the distribution $b_1(\cdot)$ is used to choose an initial action. However, where convenient, we will assume the decision maker makes some observation $o^1$ in decision epoch 1, with an arbitrary observation distribution that is independent of $s$[4]. This should be viewed as a notationally convenient way to refer to the probability distribution $b_1(s), s \in S$.

We emphasize there is subtle distinction between how actions may be implemented in a POMDP. In most POMDP applications, the decision maker chooses an action that can be used in any unobservable state, for example a robot moving in a specified direction or a doctor choosing a specific test. However, it is also possible that different actions are available in different states (see Section 3.1.3) so that the decision maker must send a signal to the controller in the unobserved system that specifies what to do in each possible state. This would correspond to specifying a decision rule in the underlying Markov decision process. For example, assuming the actions are ordered in each state, a decision rule may specify the use of the "first action" in each state. This way, the decision maker does not need to know the hidden state, but the action is still

[2]If the signal space were continuous, this equation would be replaced by a probability density or distribution.

[3]In some applications, the rewards may be observable, but must not provide any extra information about the current hidden state.

[4]This assumption is without loss of generality, since if $u_1(o|s) = P(Z_1 = o)$, then $P(X_1 = s|Z_1 = o) = P(Z_1 = o|X_1 = s)P(X_1 = s)/P(Z_1 = o) = P(X_1 = s)$.

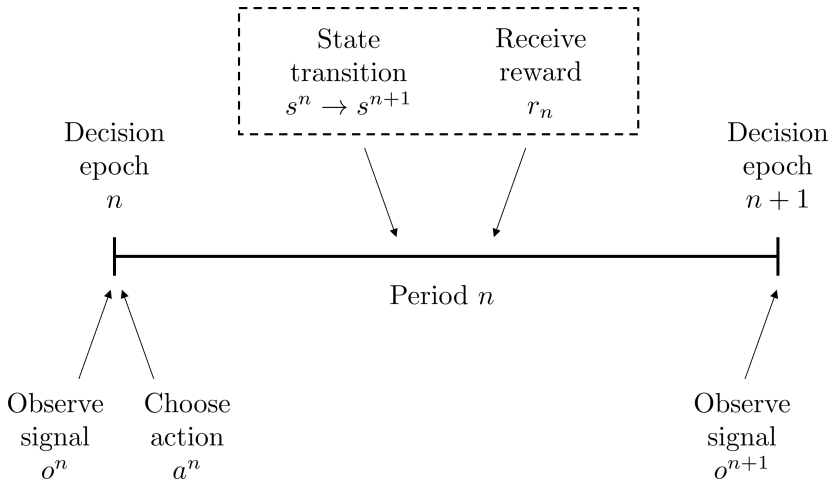


Figure 3.1: Timeline of events in period $n > 1$ of a POMDP. The decision maker observes the signal and the action choice. Events in the dashed box, namely the state transition and reward, are unobservable. Note that the first period ($n = 1$ ) begins with the decision maker's prior assessment of the state of the unobservable Markov decision process, so a signal is unnecessary.

properly specified.

In summary, a POMDP consists of an underlying[a] MDP model with unobservable states and rewards, plus a set of possible observations $O$ and a corresponding probability distribution $u(\cdot|s,a)$. While the added dynamics associated with partial observability are straightforward to understand, the mathematics of the model are considerably more complex than a fully observed MDP.

[a] Also referred to as a hidden, unobservable or core MDP in the literature.

### 3.1.2 A clinical example

We provide a high level description of a prototypical clinical decision problem to illustrate some model features. The decision is whether to treat or test a patient when the true disease state is not known. Assume the disease state can be represented by

$$S = \{none, mild, moderate, severe\}$$

and $A_s = \{do\ nothing, test, treat\}$ for each $s \in S$. If not treated the disease may progress and if treated the disease may improve. Let $p(j|s,a)$ represent the probability of change in disease state when the patient is treated or untreated (corresponding to the actions *do nothing* or *test*). This becomes a meaningful problem when the treatment is effective in controlling the disease, expensive, has serious side effects and can only be

administered once. So in essence this may be viewed as a partially observed optimal stopping problem which terminates upon treatment.

In most settings, test results will be imprecise with possible outcomes

$$O = \{negative, inconclusive, positive\},$$

which are observed with probability $u(o|s, test)$ upon testing.

In contrast to the model in the next section *each action can be applied in each state.* That is the $A_s = A$ for all $s \in S$. Moreover, the model allows for both active and passive information acquisition.

### 3.1.3 A Two-State POMDP Model

We consider a stationary version of the two-state model in Section **??**. We add a signal space $O = \{o_1, o_2\}$ and signal probabilities $u(o_1|s_1) = 0.8$, $u(o_2|s_1) = 0.2$, $u(o_1|s_1) = 0.4$ and $u(o_2|s_2) = 0.6$, which we assume to be independent of decision epoch and action choice. The specified probabilities reflect the fact that it is more likely to receive signal $o_1$ when the hidden state is $s_1$ and $o_2$ when the hidden state is $s_2$. In terms of the nomenclature above, this may be regarded as a passive model because the quality of the observation does not depend on action choice.

In addition, we assume an initial state distribution of $b_1(s_1) = P(X_1 = s_1)$ and $b_1(s_2) = P(X_1 = s_2)$. Let $\mathbf{b}_1 = (b_1(s_1), b_1(s_2))$, that is, a column vector with components $b_1(s_1)$ and $b_1(s_2)$. Since there are only two states, letting $b_1(s_1) = q$ implies $b_1(s_2) = 1 - q$ for $0 \leq q \leq 1$. Figure 3.2 provides a graphical representation of the model.

#### Computing the Value in a One-Period Model

The following calculations for a one-period model illustrate the key concepts for analysis of a POMDP. Define two deterministic Markovian decision rules, $f$ and $g$, in the underlying Markov decision process by

$$f(s) = \begin{cases} a_{1,1}, & \text{if } s = s_1 \\ a_{2,1}, & \text{if } s = s_2 \end{cases} \tag{3.2}$$

and

$$g(s) = \begin{cases} a_{1,2}, & \text{if } s = s_1 \\ a_{2,2}, & \text{if } s = s_2 \end{cases}. \tag{3.3}$$

Of course there is no need to distinguish only two decision rules. We could also include the two other deterministic Markovian decision rules in the discussion below.

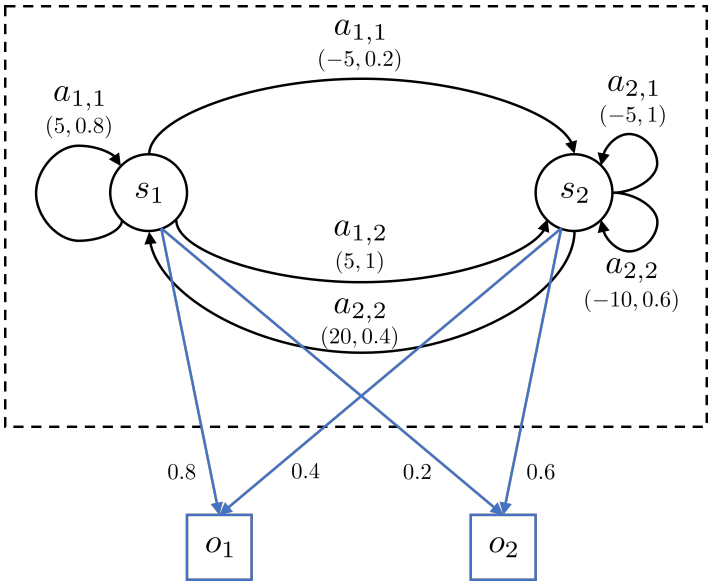


Figure 3.2: Graphical representation of the two-state POMDP model. The area in the dashed box represents the hidden (unobservable) Markov decision process. The squared boxes indicate the observable signals, with probabilities of observation on the arcs from the hidden states.

In order to implement these decision rules in the POMDP, we assume that the system is such that if the decision maker chooses either of these decision rules the controller in the unobservable system can implement them. Thus these decision become actions in the POMDP since they inform the controller which action to select. We can interpret $f$ as choosing the "first action" and $g$ as choosing the "second action", then they don't require the decision maker to have knowledge of the system state, as long as the system controller implements the action and knows what state it is in.

While a decision rule cannot depend on the hidden state, it depends on the decision maker's belief of the current system state. Since this is a one-period model, this belief is fully characterized by $\mathbf{b}_1$. While $f$ and $g$ do not depend on $\mathbf{b}_1$, the following decision rule, $d$, is an example of a decision rule with an explicit dependence on $\mathbf{b}_1$:

$$d(\mathbf{b}_1) = \begin{cases} f, & b_1(s_1) \geq \overline{q} \\ g, & b_1(s_1) < \overline{q}. \end{cases} \tag{3.4}$$

Suppose $\overline{q} = 0.5$. Then $d$ can be interpreted as choosing the decision rule $f$ if the probability of being in state $s_1$ is at least 0.5 and choosing the decision rule $g$ if the probability of being in state $s_1$ is less than 0.5. Decision rule $d$ makes clear that while decision rules are generally a function of the probability distribution of the current

state, they must still specify what to do in all hidden states.

Note that $f$ and $g$ can be thought of as pure actions, rather than decision rules, since they do not depend on anything observable to the decision maker. However, $d$ is a deterministic decision rule, since it depends on the probability $b_1(s_1)$. Clearly, $d$ is a generalization of $f$ and $g$ (through the appropriate choice of $\overline{q}$), so it should generate a reward at least as large as choosing either $f$ or $g$. The question then becomes whether there is an optimal value for $\overline{q}$, or perhaps multiple breakpoints $\overline{q}_1, \ldots, \overline{q}_k$.

To investigate, we illustrate the computation of the expected total reward under decision rules $f$ and $g$. We wish to understand under what distributions $\mathbf{b}_1$ does $f$ produce higher expected total reward. Note that even though these decision rules do not depend on $\mathbf{b}_1$, the expected total reward will depend on $\mathbf{b}_1$ since the total reward will depend on the starting state of the system. Furthermore, since this is a two-state model, a single parameter fully specifies $\mathbf{b}_1$.

Assume a terminal reward $r_2(s_1) = x$ and $r_2(s_2) = y$. Let $v_f(q)$ and $v_g(q)$ denote the expected total reward from using $f$ and $g$ when $b_1(s_1) = q = 1 - b_1(s_2)$. Then,

$$\begin{aligned} v_f(q) &= (r(s_1, a_{1,1}, s_1) + r_2(s_1))p(s_1|s_1, a_{1,1})b_1(s_1) + (r(s_1, a_{1,1}, s_2) + r_2(s_2))p(s_2|s_1, a_{1,1})b_1(s_1) \\ &\quad + (r(s_2, a_{2,1}, s_1) + r_2(s_1))p(s_1|s_2, a_{2,1})b_1(s_2) + (r(s_2, a_{2,1}, s_2) + r_2(s_2))p(s_1|s_2, a_{2,1})b_1(s_2) \\ &= (5 + x)0.8q + (-5 + y)0.2q + (0 + x)0(1 - q) + (-5 + y)1(1 - q) \\ &= (3 + 0.8x + 0.2y)q + (-5 + y)(1 - q) \\ &= -5 + y + (8 + 0.8x - 0.8y)q, \end{aligned} \tag{3.5}$$

and

$$\begin{aligned} v_g(q) &= (r(s_1, a_{1,2}, s_1) + r_2(s_1))p(s_1|s_1, a_{1,2})b_1(s_1) + (r(s_1, a_{1,2}, s_2) + r_2(s_2))p(s_2|s_1, a_{1,2})b_1(s_1) \\ &\quad + (r(s_2, a_{2,2}, s_1) + r_2(s_1))p(s_1|s_2, a_{2,2})b_1(s_2) + (r(s_2, a_{2,2}, s_2) + r_2(s_2))p(s_1|s_2, a_{2,2})b_1(s_2) \\ &= (0 + x)0q + (5 + y)1q + (20 + x)0.4(1 - q) + (-10 + y)0.6(1 - q) \\ &= (5 + y)q + (2 + 0.4x + 0.6y)(1 - q) \\ &= 2 + 0.4x + 0.6y + (3 - 0.4x + 0.4y)q. \end{aligned} \tag{3.6}$$

Figure 3.3 plots $v_f$ and $v_g$ as a function of $q$ when $x = 10$ and $y = 0$. The two functions intersect at $q = 11/17$, $v_f(q) > v_g(q)$ for $q > 11/17$ and $v_g(q) > v_f(q)$ for $q < 11/17$. Also observe that $\max\{v_f(q), v_g(q)\}$ is a convex function of $q$. We leave it as an exercise to show that for other choices of $x$ and $y$, $v_f(q)$ may lie above or below $v_g(q)$ for all values of $q$.

Some observations that will motivate our general analysis of the POMDP model:

1. We require the distribution of the state at decision epoch 2, $X_2$, to evaluate the expected total reward.

2. The expected total reward under $f$ and $g$ are linear functions of $\mathbf{b}_1$. For example, $v_f(q) = \boldsymbol{\gamma}^\mathsf{T}\mathbf{b}_1$, where $\boldsymbol{\gamma} = (3 + 0.8x + 0.2y, -5 + y)$ and $\mathbf{b}_1 = (q, 1 - q)$.

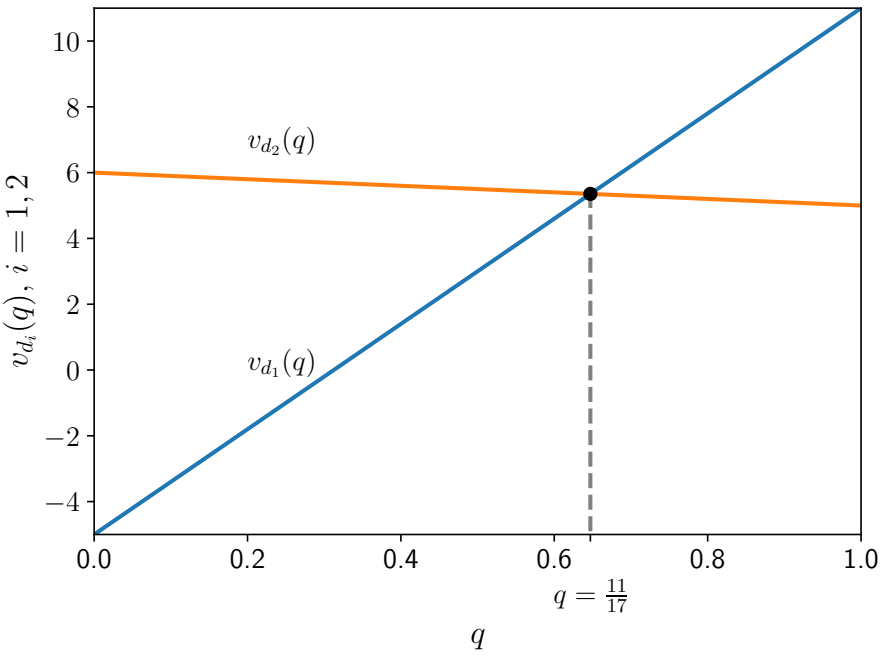


Figure 3.3: Plot of $v_f$ and $v_g$ as functions of $q$ for $q \in [0, 1]$. **(change the $v_{d_1}$ to $f$ and $v_{d_2}$ to $g$ in figure)**

3. Figure 3.3 shows that the optimal policy partitions the interval $[0, 1]$ into two non-overlapping intervals.

4. Since the decision maker cannot observe the hidden state, the choice of decision rules must be based on the probability distribution $\mathbf{b}_1$, or equivalently, $q$.

5. The above calculations do not use any information regarding the observed signal, which does not become known to the decision maker until just prior to the second decision epoch, after the transition from $X_1$ to $X_2$.

### Updating the belief probability in decision epoch 2

The probability distribution $\mathbf{b}_1$ reflects the decision maker's "belief" as to what the current state is. Next, we show that the probability distribution of the hidden state in decision epoch 2 can be easily calculated assuming we have the distribution in decision epoch 1 and the observation at decision epoch 2.

Let $b_2(s)$ be the probability that the hidden state is $s$ at decision epoch 2. Note in the following that $X_1$ is a random variable with distribution $\mathbf{b}_1$. Assuming the decision maker uses decision rule $f$ at decision epoch 1 and observes $o_1$ at decision epoch 2, we

can write $b_2(s)$ as

$$\begin{aligned} b_2(s_1) &= P(X_2 = s_1 | Z_2 = o_1, Y_1 = f(X_1), X_1) \\ &= \frac{P(Z_2 = o_1 | X_2 = s_1, Y_1 = f(X_1), X_1) P(X_2 = s_1 | Y_1 = f(X_1), X_1)}{P(Z_2 = o_1 | Y_1 = f(X_1), X_1)} \\ &= \frac{u(o_1|s_1)p(s_1|s_1, a_{1,1})b_1(s_1) + u(o_1|s_1)p(s_1|s_2, a_{2,1})b_1(s_2)}{\sum_{s \in \{s_1, s_2\}} (u(o_1|s)p(s|s_1, a_{1,1})b_1(s_1) + u(o_1|s)p(s|s_2, a_{2,1})b_1(s_2))} \\ &= \frac{0.8 \cdot 0.8q + 0.8 \cdot 0(1-q)}{(0.8 \cdot 0.8q + 0.8 \cdot 0(1-q)) + (0.4 \cdot 0.2q + 0.4 \cdot 1(1-q))} \\ &= \frac{0.64q}{0.4 + 0.32q} \end{aligned} \tag{3.7}$$

and

$$b_2(s_2) = 1 - b_2(s_1) = \frac{0.4 - 0.32q}{0.4 + 0.32q}.$$

For a concrete example, when $q = 0.2$, $b_2(s_1) = 0.128/0.464 = 0.276$. This means that after using $f$ at decision epoch 1 and observing $o_1$ at decision epoch 2, the belief that the hidden state is $s_1$ has increased from 0.2 to 0.276. Similar calculations are possible when $o_2$ is observed, or when decision rule $g$ is used instead of $f$; we leave these as an exercise for the reader. Thus, for each region in Figure 3.3, which represents either decision rule $f$ or $g$, there are two possible values of $\mathbf{b}_2$, depending on whether $o_1$ or $o_2$ are observed.

The astute reader will note that the calculations shown in (3.7) are precisely those described by *Bayes' Theorem* and that conditional on the observation in decision epoch 2, the probability distribution of the state in decision epoch 2 is a deterministic function of the probability distribution in decision epoch 1. We refer to this process as *Bayesian updating* and describe it in general in the next section.

## 3.2 The Belief State

If a decision maker chooses an action based on all the information that is available at a given epoch, including all prior signals and actions, that would be akin to using a history-dependent decision rule. However, as the process evolves, this information would grow, and the complexity associated with choosing an action may grow as well. Instead, we seek a succinct representation of this information, so that decision making in a POMDP can be done in a similar fashion as using a Markovian decision rule in a MDP, where the current state alone contains the relevant information needed to choose an action. It turns out that the information encoded in a probability distribution over the states is sufficient for decision making.

**Definition 3.1.** Let $s \in S$. Let $a^1, \ldots, a^{n-1}$ and $o^1, \ldots, o^n$ be the sequence of prior actions and signals up to time $n \geq 2$. Define the *belief state* at decision epoch $n$ to be a probability distribution defined over the state space $S$ such that $b_1(s) = P(X_1 = s)$ and

$$b_n(s) = P(X_n = s | Z_n = o^n, Y_{n-1} = a^{n-1}, \ldots, Z_2 = o^2, Y_1 = a^1, Z_1 = o^1) \quad (3.8)$$

for $n \geq 2$.

Some notes about the belief state:

1. The belief state summarizes the decision maker's beliefs about what the system state is at a given decision epoch. It is a function of past actions and observations, and not the hidden states. In other words, it is a function of only the observable parts of the process to the decision maker.

2. Although $b_n(s)$ explicitly depends on the past sequence of actions and observations, we suppress them in the notation.

3. Recall our convention that making observation $o^1$ in decision epoch 1 refers to the decision maker having the initial state distribution $b_1(s), s \in S$.

4. For each $n$ and $s \in S$, $0 \leq b_n(s) \leq 1$ and $\sum_{s \in S} b_n(s) = 1$. When $S$ is a finite set with $M$ elements, $b_n(\cdot)$ is an element of the $(M-1)$-dimensional unit simplex, $\mathcal{S}_{M-1}$, defined by

$$\mathcal{S}_{M-1} := \left\{ (x_1, \ldots, x_M) \;\middle|\; \sum_{i=1}^{M} x_i = 1, 0 \leq x_i \leq 1 \text{ for } i = 1, 2, \ldots, M \right\}. \quad (3.9)$$

5. In the two-state example above, $b_1(s) \in \mathcal{S}_1$, so it can be described by the single scalar $q \in [0, 1]$.

Later, it will be convenient to represent the belief state as an $|S|$-dimensional vector. In this case, we use $\mathbf{b}_n$ to represent the belief state in epoch $n$. We continue to use the non-bold notation when referring to individual components of $\mathbf{b}_n$.

The following result shows that the belief state in a POMDP summarizes all the relevant information needed for decision making at a given epoch. That is, no extra information about the state of the system can be obtained by considering the entire history of past actions and observations up to the current epoch. The important consequence of this result is that the belief state in a POMDP serves the same role as the state in a regular MDP.

**Theorem 3.1.** The belief state is a sufficient statistic of the past actions and observations in a POMDP. That is,

$$P(X_n = s|H_n) = P(X_n = s|Z_n = o^n, Y_{n-1} = a^{n-1}, \mathbf{b}_n), \tag{3.10}$$

where $H_n = \{Z_1 = o^1, Y_1 = a^1, \ldots, Z_n = o^n\}$ is the history up to decision epoch $n$.

*Proof.* We prove by induction. Note that by definition, $b_n(s) = P(X_n = s|H_n)$. The result is clearly true for $n = 1$, since $P(X_1 = s|\mathbf{b}_n) = P(X_1 = s) = b_1(s)$. Suppose the result is true for $n \geq 1$. Then, using Bayes' Rule, we can write $b_{n+1}(s)$ as

$$b_{n+1}(s) = P(X_{n+1} = s|Z_{n+1} = o^{n+1}, Y_n = a^n, H_n) \tag{3.11}$$

$$= \frac{P(Z_{n+1} = o^{n+1}|X_{n+1} = s, Y_n = a^n, H_n)P(X_{n+1} = s|Y_n = a^n, H_n)}{P(Z_{n+1} = o^{n+1}|Y_n = a^n, H_n)} \tag{3.12}$$

By assumption, our signal probability distribution does not depend on $H_n$, so

$$P(Z_{n+1} = o^{n+1}|X_{n+1} = s, Y_n = a^n, H_n) = u(o^{n+1}|s, a^n). \tag{3.13}$$

We can expand the term $P(X_{n+1} = s|Y_n = a^n, H_n)$ as

$$P(X_{n+1} = s|Y_n = a^n, H_n) = \sum_{j \in S} P(X_{n+1} = s|X_n = j, Y_n = a^n, H_n)P(X_n = j|Y_n = a^n, H_n) \tag{3.14}$$

$$= \sum_{j \in S} p(s|j, a^n)b_n(j), \tag{3.15}$$

where the first summand is the transition probability, which does not depend on the history, and the second is the probability distribution of the hidden state at epoch $n$. Note that the second term is conditioned on $Y_n$, which is irrelevant since $Y_n$ is not yet observed at $X_n$. So $P(X_n = j|Y_n = a^n, H_n) = P(X_n = j|H_n) = b_n(j)$, by definition. Putting it all together, we have

$$b_{n+1}(s) = \frac{u(o^{n+1}|s, a^n) \sum_{j \in S} p(s|j, a^n)b_n(j)}{\sum_{s' \in S} u(o^{n+1}|s', a^n) \sum_{j \in S} p(s'|j, a^n)b_n(j)}. \tag{3.16}$$

Thus, computing $b_{n+1}(s)$ requires only the most recent observation, action, and belief state, as desired. □

Equation (3.16) is recursive equation defining the belief state. Analogous to a state transition in an MDP, where the transition probability depends only on the current state, the transition of the belief state depends only on the current belief state.

Given this result, we revise our description of the POMDP dynamics to include the belief state. After observing signal $o^n$ in epoch $n$, the decision maker updates their

belief about the true state of the system, and then an action is chosen. The rest of the dynamics remain unchanged. For $n = 1$, we assume the signal $o^1$ is artificial and the decision maker possesses the distribution $b_1(s)$ a priori. Figure 3.4 illustrates the inclusion of the belief state in the timing of events in one period.

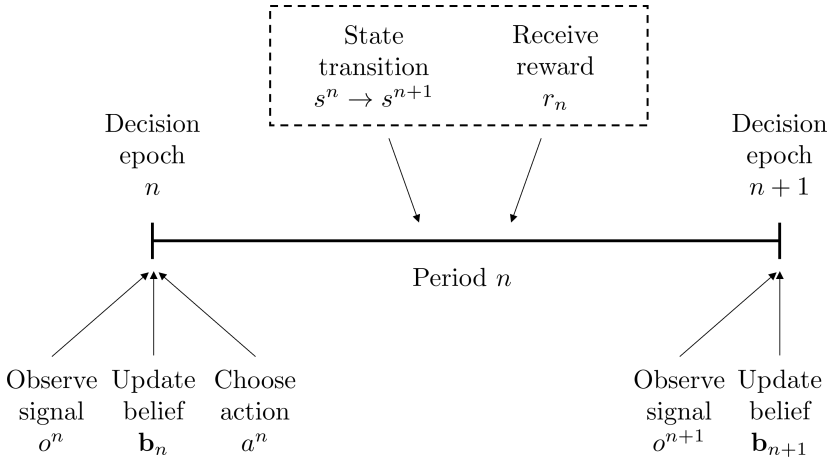


Figure 3.4: Timeline for a POMDP in one period. The decision maker observes the signal, the updated belief, and the action choice. Events in the dashed box, namely the state transition and reward, are unobservable.

### 3.2.1 Belief State Updating

The proof of Theorem 3.1 shows that given a belief state in decision epoch $n$, it is straightforward to update the belief state in decision epoch $n + 1$:

$$b_{n+1}(s) = \frac{u(o^{n+1}|s, a^n) \sum_{j \in S} p(s|j, a^n) b_n(j)}{\sum_{s' \in S} u(o^{n+1}|s', a^n) \sum_{j \in S} p(s'|j, a^n) b_n(j)}. \tag{3.17}$$

Equation (3.17) shows that the updated belief state is the result of the following four-step process.

1. The decision maker chooses action $a^n$ on the basis of the belief state $b_n$ at the decision epoch $n$.

2. The hidden process moves to state $s$ according to the probability distribution $\sum_{j \in S} p(s|j, a^n) b_n(j)$.

3. The system generates the signal $o^{n+1}$ based on the probability distribution $u(o^{n+1}|s, a^n)$.

4. The decision maker updates the belief state to $b_{n+1}$ using (3.17).

Our reason for spelling out this updating process in detail is to emphasize that steps 2 and 3 are probabilistic, while step 4 is deterministic.

Observe that the numerator of (3.17) equals the probability that the hidden state is $s$ and that the decision maker observes $o^{n+1}$, while the denominator equals the probability that the decision maker observes $o^{n+1}$, independent of $s$. These computations also show that $b_{n+1}(s)$ takes into account all the information available to the decision maker at decision epoch $n+1$, through $b_n(\cdot)$, the action at decision epoch $n$ and the observation $o^{n+1}$ at decision epoch $n+1$.

## 3.3 MDP formulation of a POMDP

In this section, we demonstrate that a POMDP with a finite state space can be equivalently represented as an MDP with a continuous state space, namely, the space of belief states. To evaluate policies and find optimal policies in a POMDP, we can convert it into an MDP in which the belief state of the POMDP becomes the state of the MDP. Accordingly, we must redefine the action sets, rewards, and transition probabilities in terms of the belief states. We emphasize that we will refer to two Markov decision processes, the original one underlying the POMDP model, and the equivalent MDP formulation of the POMDP model. We refer to the former as the *underlying Markov decision process* and the latter as the *derived Markov decision process*. **We use a tilde to distinguish entities in the derived Markov decision process from those in the underlying Markov decision process.** For example, the derived state space will be $\tilde{S}$. Section **(XXX)** provides a relevant summary of Markov decision processes with continuous state spaces, including solution algorithms. **(need a mini section on continuous state finite action MDP?)**

We assume $S$ and $O$ are discrete and finite. We write them as $S = \{s_1, \ldots, s_M\}$ and $O = \{o_1, \ldots, o_K\}$.

### 3.3.1 Model components

Figure 3.4 displays the underlying dynamics of a POMDP. At decision epoch $n$, the decision maker computes the belief state $\mathbf{b}_n$ after observing signal $o^n$, chooses action $a^n$, then prior to decision epoch $n+1$ the system changes state from $s^n$ to $s^{n+1}$ and generates a signal $o^{n+1}$, which the decision maker uses to update the belief state to $\mathbf{b}_{n+1}$.

**Decision Epochs:** Decision epochs correspond to the point in time immediately after the decision maker updates the belief state. The model may be either finite or infinite horizon.

$$\tilde{T} = \{1, 2, \ldots, N\}, \quad N \leq \infty.$$

**States:** Assuming $S$ has $M$ elements,

$$\tilde{S} = \mathcal{S}_{M-1}.$$

The states of the derived Markov decision process correspond to the belief states. Thus, $\tilde{S}$ is an uncountable, compact set even when $S$ is finite.

**Actions:** Since the decision maker does not observe the hidden state, in general, an action in the POMDP must correspond to a decision rule of the underlying Markov decision process, so that it can be implemented in each hidden state. In other words, an action in the derived MDP must be a vector of length $M$ that specifies an action for each hidden state. Thus,

$$\tilde{A} \subseteq \prod_{s \in S} A_s = \{(a(s_1), a(s_2), \ldots, a(s_M)) \mid a(s_m) \in A_{s_m}, m = 1, 2, \ldots, M\}.$$

We define $\tilde{A}$ as a subset of $\Pi_{s \in S} A_s$ to emphasize that it may only contain a subset of all possible actions in the underlying Markov decision process. Since $\tilde{A}$ consists of vectors of length $M$, we will write actions in the derived MDP as the vector $\mathbf{a}$, with components $a(s)$ for each $s \in S$. The two-state model in Section 3.1.3 provides an example where an action in the POMDP corresponds to a (Markovian deterministic) decision rule in the underlying MDP.

Note that most practical examples are compatible with the assumption that the available actions in each state are the same. In other words, $A_s = A$ for all $s \in S$, and $\tilde{A} = A^M$. In these examples, it is common to simplify the action set to $\tilde{A} = A$. That is, an action in the POMDP corresponds to an action in the underlying MDP. This scenario is applicable when the decision maker, rather than the system, is the one implementing the action and thus must choose a single action that is implementable no matter the hidden state. This case is illustrated in the inspection problem in Section 3.4.1. Of course, the decision maker can still choose an action using a decision rule that depends on the belief state.

Our development in this chapter will continue with the general case where actions in the POMDP are decision rules in the MDP. However, we will highlight examples where $A_s = A$ for all $s \in S$ to illustrate the simpler formulation.

**Rewards:** Since the reward in the underlying MDP, $r(s, a, j)$, is a function of hidden states $s$ and $j$, we convert it to a function of the belief state $\mathbf{b}$ as follows. Define $\tilde{r}(\mathbf{b}, \mathbf{a})$ for $\mathbf{b} \in \tilde{S}$ and $\mathbf{a} \in \tilde{A}$ by

$$\tilde{r}(\mathbf{b}, \mathbf{a}) = \sum_{s \in S} \sum_{j \in S} r(s, a(s), j) p(j|s, a(s)) b(s). \tag{3.18}$$

In the above expression, the inner sum (over $j$) gives the expected reward conditional on the hidden system occupying state $s$ and choosing action $a(s)$. The outer sum (over

$s$) takes into account that $s$ is unobservable and that the decision maker enters the decision epoch with the belief state $\mathbf{b}$.

When the underlying reward function is independent of $j$, (3.18) simplifies to

$$\tilde{r}(\mathbf{b}, \mathbf{a}) = \sum_{s \in S} r(s, a(s)) b(s). \tag{3.19}$$

Equations (3.18) and (3.19) show that $\tilde{r}(\mathbf{b}, \mathbf{a})$ is **linear** in $\mathbf{b}$. This observation will play a key role in subsequent computations. Note also that even when the underlying reward depends on the subsequent state of the hidden process as in (3.18), $\tilde{r}(\mathbf{b}, \mathbf{a})$ depends only on the belief state at the current decision epoch and not on the belief state at the subsequent decision epoch.

For $\mathbf{b} \in \tilde{S}$, the terminal reward in a finite horizon model becomes

$$\tilde{r}_N(\mathbf{b}) = \sum_{s \in S} r_N(s) b(s). \tag{3.20}$$

**Transition Probabilities:** We begin with some additional notation to simplify some expressions. For each $\mathbf{b} \in \tilde{S}$ and $\mathbf{a} \in \tilde{A}$, define the probability of observing $o \in O$ by

$$\eta(o|\mathbf{b}, \mathbf{a}) := \sum_{s' \in S} \sum_{s \in S} u(o|s', a(s)) p(s'|s, a(s)) b(s). \tag{3.21}$$

Let the updated belief state be represented by $\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})$, which is a vector of dimension $M$ and depends on $o$, $\mathbf{a}$, and $\mathbf{b}$. We write the $j$-th component of the updated belief state when observing $o \in O$ as

$$\tau(o, \mathbf{a}, \mathbf{b})(j) := \frac{\sum_{s \in S} u(o|j, a(s)) p(j|s, a(s)) b(s)}{\sum_{s' \in S} \sum_{s \in S} u(o|s', a(s)) p(s'|s, a(s)) b(s)} = \frac{\sum_{s \in S} u(o|j, a(s)) p(j|s, a(s)) b(s)}{\eta(o|\mathbf{b}, \mathbf{a})}. \tag{3.22}$$

Then

$$\tilde{p}(\mathbf{b}'|\mathbf{b}, \mathbf{a}) = \begin{cases} \eta(o|\mathbf{b}, \mathbf{a}) & \text{when } b'(j) = \tau(o, \mathbf{a}, \mathbf{b})(j) \text{ for all } j \in S \text{ and for each } o \in O \\ 0 & \text{otherwise.} \end{cases} \tag{3.23}$$

Recall from the belief state updating equation (3.16) that the next belief state is a deterministic function of the current belief state and signal. Hence, the probability of observing a particular belief state is equal to the probability of observing the specific signal that leads to that belief state. Observe that when $O$ has $K$ elements, $\tilde{p}(\mathbf{b}'|\mathbf{b}, \mathbf{a})$ assumes at most $K$ non-zero values, at most one for each $o \in O$.

### Matrix Notation

Much of the literature represents the rewards and transition probabilities using matrix notation, which simplifies calculations and makes the dimensions of various objects more apparent. We present these quantities below.

- $\mathbf{b}$: column vector with components $b(s_1), \ldots, b(s_M)$
- $\mathbf{P_a}$: $M \times M$ matrix with its $(s,j)$-th component equal to $p(j|s,a(s))$[5]
- $\mathbf{R_a}$: $M \times M$ matrix with $(s,j)$-th entry $r(s,a(s),j)$
- $\mathbf{r_a}$: $M$-dimensional column vector with components $r(s,a(s))$ for all $s \in S$
- $\mathbf{r}_N$: $M$-dimensional column vector with components $r_N(s)$ for all $s \in S$ (for the finite horizon setting)
- $\mathbf{U}^o_\mathbf{a}$: $M \times M$ **diagonal** matrix with entries $u(o|s_1,a(s_1)), \ldots, u(o|s_M,a(s_M))$
- $\mathbf{e}$: $M$-component column vector with all entries equal to 1.

When the reward function of the underlying MDP is $r(s,a,j)$, then from (3.18)

$$\tilde{r}(\mathbf{b},\mathbf{a}) = \text{diag}(\mathbf{R_a}\mathbf{P_a}^\mathsf{T})\mathbf{b}, \tag{3.24}$$

where $\text{diag}(\mathbf{R_a}\mathbf{P_a}^\mathsf{T})$ denotes an $M \times M$ diagonal matrix containing the diagonal elements of $\mathbf{R_a}\mathbf{P_a}^\mathsf{T}$, i.e., the elements $(\mathbf{R_a}\mathbf{P_a}^\mathsf{T})_{j,j}$ for $j = 1, \ldots, M$. When the underlying reward function $r(s,a)$ does not depend on $j$, then from (3.19)

$$\tilde{r}(\mathbf{b},\mathbf{a}) = \mathbf{r_a}^\mathsf{T}\mathbf{b}. \tag{3.25}$$

For the quantities used to write the transition probabilities, we have

$$\eta(o|\mathbf{b},\mathbf{a}) = \mathbf{e}^\mathsf{T}\mathbf{U}^o_\mathbf{a}\mathbf{P_a}^\mathsf{T}\mathbf{b} \quad \text{and} \quad \boldsymbol{\tau}(o,\mathbf{a},\mathbf{b}) = \frac{\mathbf{U}^o_\mathbf{a}\mathbf{P_a}^\mathsf{T}\mathbf{b}}{\eta(o|\mathbf{b},\mathbf{a})} \tag{3.26}$$

In this notation

$$\tilde{p}(\mathbf{b}'|\mathbf{b},\mathbf{a}) = \begin{cases} \eta(o|\mathbf{b},\mathbf{a}) & \text{when } \mathbf{b}' = \boldsymbol{\tau}(o,\mathbf{a},\mathbf{b}) \text{ for each } o \in O \\ 0 & \text{otherwise} \end{cases} \tag{3.27}$$

Equation (3.27) shows that the only values for $\mathbf{b}'$ with non-zero probability are those represented by $\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})$ for $o \in O$. When $O$ has $K$ elements, there are at most $K$ distinct vectors $\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})$.

### 3.3.2 The Two-State Model Revisited

We now illustrate the derived MDP model for the two-state POMDP.

**Decision Epochs:**

$$\tilde{T} = \{1, 2, \ldots, N\}, \quad N \leq \infty$$

[5]With this definition, we are implicitly assuming deterministic decision rules in the derived MDP

**States:**

$$\tilde{S} = \{(q_1, q_2) \mid q_1 + q_2 = 1, q_1 \geq 0, q_2 \geq 0\} = \mathcal{S}_1$$

The $q_i$ represents the probability that the hidden state is $s_i, i = 1, 2$. Note that the state space can be equivalently represented as

$$\{q \mid 0 \leq q \leq 1\} = [0, 1],$$

where $q$ is the probability that the hidden state is $s_1$, as shown in Section 3.1.3. Below, we use $\mathcal{S}_1$ to emphasize the fact that the belief state is a probability distribution with two components.

**Actions:** We consider an action set corresponding to the two decision rules defined in equations (3.2) and (3.3).

$$\tilde{A} = \{(a_{1,1}, a_{2,1}), (a_{1,2}, a_{2,2})\} = \{f, g\}.$$

The first component of each action in $\tilde{A}$ denotes the action to choose when the hidden process is in state $s_1$ and the second component denotes the action to choose if the hidden process is in state $s_2$. Note that this $\tilde{A}$ is a proper subset of the set of Markovian deterministic decision rules in the underlying two-state Markov decision process. That is, we have excluded $(a_{1,1}, a_{2,2})$ and $(a_{1,2}, a_{2,1})$.

**Rewards:** We derive the reward when the decision maker chooses action $(a_{1,1}, a_{2,1}) \in \tilde{A}$ after computing $\mathbf{b} = (q_1, q_2) \in \tilde{S}$ to be

$$\begin{aligned}
\tilde{r}((q_1, q_2), (a_{1,1}, a_{2,1})) &= (r(s_1, a_{1,1}, s_1)p(s_1|s_1, a_{1,1}) + r(s_1, a_{1,1}, s_2)p(s_2|s_1, a_{1,1}))q_1 \\
&\quad + (r(s_2, a_{2,1}, s_1)p(s_1|s_2, a_{2,1}) + r(s_2, a_{2,1}, s_2)p(s_2|s_2, a_{2,1}))q_2 \\
&= (5 \cdot 0.8 + (-5)0.2)q_1 + (0 \cdot 0 + (-5)1)q_2 \qquad (3.28) \\
&= 3q_1 - 5q_2
\end{aligned}$$

Observe that the reward is a linear function of the vector $(q_1, q_2)$. Since $q_2 = 1 - q_1$ the reward reduces to $-5 + 8q_1$. We leave it to the reader the derive $\tilde{r}((q_1, q_2), (a_{1,2}, a_{2,2}))$.

**Transition Probabilities:** We derive the probability that the hidden state occupies $s_1$ when the decision maker chooses action $(a_{1,1}, a_{2,1}) \in \tilde{A}$ given belief state $\mathbf{b} = (q_1, q_2) \in \tilde{S}$. From (3.21), the probability of observing $o_1$ is given by

$$\begin{aligned}
\eta(o_1|(q_1, q_2), (a_{1,1}, a_{2,1})) &= u(o_1|s_1)(p(s_1|s_1, a_{1,1})q_1 + p(s_1|s_2, a_{2,1})q_2) \\
&\quad + u(o_1|s_2)(p(s_2|s_1, a_{1,1})q_1 + p(s_2|s_2, a_{2,1})q_2) \\
&= 0.8(0.8q_1 + 0q_2) + 0.4(0.2q_1 + 1q_2) = 0.72q_1 + 0.4q_2
\end{aligned}$$

and the probability of observing $o_2$ equals $0.28q_1 + 0.6q_2$. Then, from (3.22), the probability the hidden state equals $s_1$ satisfies

$$\begin{aligned}\tau(o_1,(a_{1,1},a_{2,1}),(q_1,q_2))(s_1) &= \frac{u(o_1|s_1)(p(s_1|s_1,a_{1,1})q_1 + p(s_1|s_2,a_{2,1})q_2)}{\eta(o_1(q_1,q_2),(a_{1,1},a_{2,2}))} \\ &= \frac{0.8(0.8q_1+0q_2)}{0.72q_1+0.4q_2} = \frac{0.64q_1}{0.72q_1+0.4q_2}. \end{aligned} \tag{3.29}$$

The probability the hidden state equals $s_2$ equals $1 - \tau(o_1,(a_{1,1},a_{2,1}),(q_1,q_2))(s_1)$, is given by

$$\tau(o_1,(a_{1,1},a_{2,1}),(q_1,q_2))(s_2) = \frac{0.08q_1+0.4q_2}{0.72q_1+0.4q_2}. \tag{3.30}$$

Hence,

$$\boldsymbol{\tau}(o_1,(a_{1,1},a_{2,1}),(q_1,q_2)) = \left(\frac{0.64q_1}{0.72q_1+0.4q_2}, \frac{0.08q_1+0.4q_2}{0.72q_1+0.4q_2}\right) \tag{3.31}$$

Similar calculations show that

$$\boldsymbol{\tau}(o_2,(a_{1,1},a_{2,1}),(q_1,q_2)) = \left(\frac{0.16q_1}{0.28q_1+0.6q_2}, \frac{0.12q_1+0.6q_2}{0.28q_1+0.6q_2}\right) \tag{3.32}$$

Putting it all together, the transition probabilities in the derived model satisfy

$$\tilde{p}((q_1',q_2')|(q_1,q_2),(a_{1,1},a_{2,1})) = \begin{cases} 0.72q_1+0.4q_2 & \text{for } (q_1',q_2') = \boldsymbol{\tau}(o_1,(a_{1,1},a_{2,1}),(q_1,q_2)) \\ 0.28q_1+0.6q_2 & \text{for } (q_1',q_2') = \boldsymbol{\tau}(o_2,(a_{1,1},a_{2,1}),(q_1,q_2)) \\ 0 & \text{otherwise,} \end{cases} \tag{3.33}$$

so that $\tilde{p}(\cdot|(p_1,p_2),(a_{1,1},a_{2,1}))$ is indeed a transition probability. It only takes two positive values, one for each observation.

Numerically, suppose that the belief state is $(0.2, 0.8)$, that the decision maker chooses action $(a_{1,1}, a_{2,1})$ and observes $o_1$. Then the subsequent belief state, given by (3.31), equals $(0.276, 0.724)$, which is observed by the decision maker with probability 0.464. We leave it to the reader to compute the other possible value of the belief state, which occurs when the decision maker observes $o_2$.

We now display the matrix representation for action $\mathbf{a} := (a_{1,1}, a_{2,1})$ in the two-state model. We leave it as exercise to provide it for $(a_{1,2}, a_{2,2})$.

$$\mathbf{P}_{\mathbf{a}} = \begin{bmatrix} 0.8 & 0.2 \\ 0 & 1 \end{bmatrix}, \ \mathbf{U}_{\mathbf{a}}^{o_1} = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.4 \end{bmatrix}, \ \mathbf{U}_{\mathbf{a}}^{o_2} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.6 \end{bmatrix}, \ \mathbf{R}_{\mathbf{a}} = \begin{bmatrix} 5 & -5 \\ 0 & -5 \end{bmatrix} \tag{3.34}$$

Using these quantities, we leave it to the reader to compute $\eta(o|\mathbf{b},\mathbf{a})$ and $\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})$ from (3.26).

As noted above, the observation process in the two-state POMDP model is passive. Suppose there is a costly action $\bar{\mathbf{a}}$ defined by

$$\mathbf{P}_{\bar{\mathbf{a}}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \ \mathbf{U}_{\bar{\mathbf{a}}}^{o_1} = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.05 \end{bmatrix}, \ \mathbf{U}_{\bar{\mathbf{a}}}^{o_2} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.95 \end{bmatrix}, \ \mathbf{r}_{\bar{\mathbf{a}}} = \begin{bmatrix} -15 \\ -20 \end{bmatrix} \tag{3.35}$$

The presence of this action adds an information seeking possibility. We encourage the reader to verbally interpret these model components.

## 3.4 Examples

In this section, we formulate several applications as POMDPs. In contrast to an MDP formulation, a POMDP formulation requires the additional specification of the observations. We also provide derived MDP formulations for several examples.

### 3.4.1 Preventive Maintenance and Inspection

A machine that manufactures widgets can be in one of two conditions, "good" or "bad". At any decision epoch, the decision maker can do one of three things: 1) minimal observation of the output, 2) actively inspect the output or 3) repair the machine. As a consequence of choosing an action, the decision maker receives a signal regarding the status of the equipment. Assume all actions each require one period and generate a signal at the end of the period. The first two actions provide a signal of the machine's health with the active inspection providing a more accurate signal; neither affects machine state. The repair action will result in the machine being in the "good" state by the start of the next period. Different costs are associated with observing and repairing the machine. Revenue depends on the quality of the produced widgets, and thus depends on the machine state. We regard this model as one with both passive and information seeking actions. The first action is passive and amounts to a visual inspection of the widgets. The second action is information seeking and could come in the form of a diagnostic of the widget's functionality. A POMDP formulation follows.

**Decision Epochs:** The horizon may be either finite or infinite.

$$T = \{1, 2, \ldots, N\}, \quad N \leq \infty$$

**Hidden States:**

$$S = \{s_1, s_2\}$$

where $s_1$ = "good" and $s_2$ = "bad".

**Actions:**

$$A = \{a_1, a_2, a_3\}$$

where $a_1$ = "do nothing", $a_2$ = "inspect" and $a_3$ = "repair".

**Observations:**

$$O = \{o_1, o_2\}$$

where $o_1$ = "normal" and $o_2$ = "abnormal". The probability of these observations depends on the hidden state and the action at the previous decision epoch. Since there are only two signals, we need only specify the probability of observing $o_1$:

$$u(o_1|s,a) = \begin{cases} \alpha_1 & s = s_1, a = a_1 \\ \alpha_2 & s = s_2, a = a_1 \\ \beta_1 & s = s_1, a = a_2 \\ \beta_2 & s = s_2, a = a_2 \\ 1 & s \in \{s_1, s_2\}, a = a_3 \end{cases}$$

The expression for $u(o_1|s,a)$ assumes that it is more likely to observe a "normal" signal in the "good" state than the "bad" state ($\alpha_1 > \alpha_2$, $\beta_1 > \beta_2$) and that inspection provides more accurate information than observation ($\beta_1 > \alpha_1$, $\beta_2 > \alpha_2$). If the decision maker decides to repair the equipment, it takes one period, the hidden state becomes "good" at the next decision epoch and the observed signal will be "normal" with probability 1.

**Rewards:** Rewards depend on assumptions regarding when the equipment changes state between decision epochs. We assume that a transition between states occurs at the end of a period, immediately preceding the subsequent decision epoch. Under this assumption, rewards are accrued prior to, and thus are independent of, the subsequent state. The reward function is given by

$$r(s,a) = \begin{cases} r_1 & s = s_1, a = a_1 \\ r_2 & s = s_2, a = a_1 \\ -k & s \in \{s_1, s_2\}, a = a_2 \\ -K & s \in \{s_1, s_2\}, a = a_3. \end{cases}$$

We assume that the equipment generates greater revenue in the "good" state ($r_1 > r_2$) and that it costs more to repair than to inspect ($K > k$).

**Transition Probabilities:**

$$p(s'|s,a) = \begin{cases} 1-\gamma & s' = s_1, s = s_1, a \in \{a_1, a_2\} \\ \gamma & s' = s_2, s = s_1, a \in \{a_1, a_2\} \\ 0 & s' = s_1, s = s_2, a \in \{a_1, a_2\} \\ 1 & s' = s_2, s = s_2, a \in \{a_1, a_2\} \\ 1 & s' = s_1, s \in \{s_1, s_2\}, a = a_3 \\ 0 & s' = s_2, s \in \{s_1, s_2\}, a = a_3 \end{cases} \tag{3.36}$$

When the decision maker decides to observe or inspect, (3.36) shows that the probability that the equipment changes status from "good" to "bad" in one period equals $\gamma$ and that it remains in the "bad" state with certainty once it reaches there. When the decision maker decides to repair, the equipment moves to the "good" state with certainty.

We depict the dynamics graphically in Figure **(xxx)**.

**(insert figure)**

Next, we formulate the derived MDP for this example. Decision epochs remain unchanged.

**States:** Since there are two hidden states,

$$\tilde{S} = \{(q_1, q_2) \mid q_1 + q_2 = 1, q_1 \geq 0, q_2 \geq 0\} = \mathcal{S}_1,$$

where $q_1$ is the probability the system is in the "good" state and $q_2$ is the probability the system is in the "bad" state.

**Actions:** In this example, the action set is the same for each hidden state and the decision maker is the one implementing the actions. Thus, in the derived MDP, the action set simplifies to

$$\tilde{A} = A = \{a_1, a_2, a_3\},$$

where $a_1, a_2, a_3$ are as defined in the POMDP.

**Rewards:** Let $\mathbf{b} = (q_1, q_2)$. From equation (3.19), with a scalar action,

$$\tilde{r}(\mathbf{b}, a) = \sum_{s \in S} r(s, a) b(s) = \begin{cases} r_1 q_1 + r_2 q_2 & a = a_1 \\ -k & a = a_2 \\ -K & a = a_3 \end{cases}$$

**Transition Probabilities:** Let $\mathbf{b} = (q_1, q_2)$ and $\mathbf{b}' = (q_1', q_2')$. From equation (3.21),

$$\begin{aligned} \eta(o_1|\mathbf{b}, a) &= \sum_{s' \in S} u(o_1|s', a) \sum_{s \in S} p(s'|s, a) b(s) \\ &= \begin{cases} \alpha_1(1-\gamma)q_1 + \alpha_2(\gamma q_1 + q_2) & a = a_1 \\ \beta_1(1-\gamma)q_1 + \beta_2(\gamma q_1 + q_2) & a = a_2 \\ 1 & a = a_3 \end{cases} \end{aligned}$$

and

$$\begin{aligned} \eta(o_2|\mathbf{b}, a) &= \sum_{s' \in S} u(o_2|s', a) \sum_{s \in S} p(s'|s, a) b(s) \\ &= \begin{cases} (1-\alpha_1)(1-\gamma)q_1 + (1-\alpha_2)(\gamma q_1 + q_2) & a = a_1 \\ (1-\beta_1)(1-\gamma)q_1 + (1-\beta_2)(\gamma q_1 + q_2) & a = a_2 \\ 0 & a = a_3 \end{cases} \end{aligned}$$

From equation (3.22),

$$\boldsymbol{\tau}(o_1, a_1, \mathbf{b}) = \left( \frac{\alpha_1(1-\gamma)q_1}{\alpha_1(1-\gamma)q_1 + \alpha_2(\gamma q_1 + q_2)}, \frac{\alpha_2(\gamma q_1 + q_2)}{\alpha_1(1-\gamma)q_1 + \alpha_2(\gamma q_1 + q_2)} \right) \tag{3.37}$$

$$\boldsymbol{\tau}(o_1, a_2, \mathbf{b}) = \left( \frac{\beta_1(1-\gamma)q_1}{\beta_1(1-\gamma)q_1 + \beta_2(\gamma q_1 + q_2)}, \frac{\beta_2(\gamma q_1 + q_2)}{\beta_1(1-\gamma)q_1 + \beta_2(\gamma q_1 + q_2)} \right) \tag{3.38}$$

$$\boldsymbol{\tau}(o_1, a_3, \mathbf{b}) = (1, 0) \tag{3.39}$$

$$\boldsymbol{\tau}(o_2, a_1, \mathbf{b}) = \left( \frac{(1-\alpha_1)(1-\gamma)q_1}{(1-\alpha_1)(1-\gamma)q_1 + (1-\alpha_2)(\gamma q_1 + q_2)}, \frac{(1-\alpha_2)(\gamma q_1 + q_2)}{(1-\alpha_1)(1-\gamma)q_1 + (1-\alpha_2)(\gamma q_1 + q_2)} \right) \tag{3.40}$$

$$\boldsymbol{\tau}(o_2, a_2, \mathbf{b}) = \left( \frac{(1-\beta_1)(1-\gamma)q_1}{(1-\beta_1)(1-\gamma)q_1 + (1-\beta_2)(\gamma q_1 + q_2)}, \frac{(1-\beta_2)(\gamma q_1 + q_2)}{(1-\beta_1)(1-\gamma)q_1 + (1-\beta_2)(\gamma q_1 + q_2)} \right) \tag{3.41}$$

Note that $\boldsymbol{\tau}(o_2, a_3, \mathbf{b})$ is not well-defined, since $\eta(o_2|\mathbf{b}, a_3) = 0$. Under $a_3$, with probability 1 the system transitions to $s_1$ and observation $o_1$ is generated. This is an example where there are fewer updated belief states than there are observations, for a given current belief state and action.

Finally,

$$\tilde{p}(\mathbf{b}'|\mathbf{b}, a) = \begin{cases} \eta(o_1|\mathbf{b}, a_1) & \mathbf{b}' = \boldsymbol{\tau}(o_1, a_1, \mathbf{b}) \\ \eta(o_1|\mathbf{b}, a_2) & \mathbf{b}' = \boldsymbol{\tau}(o_1, a_2, \mathbf{b}) \\ \eta(o_1|\mathbf{b}, a_3) & \mathbf{b}' = \boldsymbol{\tau}(o_1, a_3, \mathbf{b}) \\ \eta(o_2|\mathbf{b}, a_1) & \mathbf{b}' = \boldsymbol{\tau}(o_2, a_1, \mathbf{b}) \\ \eta(o_2|\mathbf{b}, a_2) & \mathbf{b}' = \boldsymbol{\tau}(o_2, a_2, \mathbf{b}) \\ 0 & \text{otherwise} \end{cases}$$

### 3.4.2 Models with Unknown Parameters - Bayesian Decision Problems

Models with unknown parameters represent a particularly useful class of POMDPs. In such models the decision maker knows the form of the distribution of a random variable that affects rewards and transition probabilities up to the unknown parameter values. For example, in a production process the defect rate may be binomial with unknown defect probability. The decision maker acquires information about the parameter through a sequence of actions that affect both the information received and the

reward earned. We illustrate how to formulate such models through examples. The key feature is that the state represents the unknown parameters and remains unchanged over time.

### Newsvendor with Unknown Demand

As an example, we formulate a newsvendor model in which the demand is Poisson distributed with unknown parameter $\mu$. We denote the Poisson probability mass function by $f(d|\mu)$. We assume that $\mu$ can assume a finite number of values $\{\mu_1, \mu_2 \ldots, \mu_M\}$ and the newsvendor has a prior distribution on $\mu$, denoted $w_1(\mu)$.

In such a model, inventory is perishable and lasts only one period. Each period, the decision maker purchases units of a product from a supplier at a cost $c$ and sells them at price $b_H > c$ throughout the period. If any items remain unsold at the end of the period, they are scrapped at a reduced price $b_L < c$. Although on the surface, this appears to be a one-period model, decision epochs are linked through changes in acquired knowledge regarding the value of the unknown parameter.

**Decision Epochs:** The model may be either finite or infinite horizon.

$$T = \{1, 2, \ldots, N\}, \quad N \leq \infty$$

**Hidden States:** States represent possible values for the unknown parameter. Under the above assumption,

$$S = \{\mu_1, \mu_2 \ldots, \mu_M\}$$

In greater generality, $\mu$ could take values in a continuum.

**Actions:** Actions represent the number of units to order at each decision epoch,

$$A = \{1, \ldots, L\},$$

where $L$ is some large finite value representing the maximum amount that can be ordered per period.

**Observations:** The newsvendor observes only the number of units sold. If external demand is less than the quantity ordered, the newsvendor observes demand, but when demand exceeds inventory, demand is *censored* by the order quantity. That is, the decision maker is only guaranteed to observe a lower bound on the true demand. Thus,

$$O = \{0, 1, \ldots, L\}.$$

It is important to note that action choice determines the values of $o$ that have positive probability. We can write the observation distribution as

$$u(o|\mu, a) = \begin{cases} f(o|\mu) & o < a, \mu \in S \\ \sum_{i=a}^{\infty} f(i|\mu) & o = a, \mu \in S \\ 0 & o > a \end{cases}$$

where we represent the state by $\mu$.

**Rewards:** The newsvendor receives rewards that depend on the values of $a$ and $\mu$. To account for the effect of the unknown parameter, we use the following expression for the *expected* reward

$$r(\mu, a) = \sum_{d=0}^{a-1}((b_H - c)d + (b_L - c)(a - d))f(d|\mu) + \sum_{d=a}^{\infty}(b_H - c)af(d|\mu). \tag{3.42}$$

When the order quantity $a$ exceeds demand $d$, the first expression on the right hand side of (3.42) combines the gain from sales at a profit $p_H - c$ with the loss of $p_L - c$ for unsold items when the order quantity $a$ exceeds demand $d$. The second expression corresponds to when the demand equals or exceeds the order quantity.

**Transition Probabilities:** Since the parameter value remains constant over the planning horizon, for all $a \in A$

$$p(j|s, a) = \begin{cases} 1 & j = s \\ 0 & j \neq s. \end{cases}$$

Some features of this model include:

1. The state of the hidden process does not change.
2. The decision maker must trade-off reward with knowledge gained by incurring losses.
3. A continuous version of this model with known distribution function $F(d)$ has a closed form optimal solution

$$a^* = F\left(\frac{b_H - c}{b_H - b_L}\right) \tag{3.43}$$

The quantity $a^*$ is sometimes referred to as the *critical fractile* and represents a percentile of the demand distribution. It represents the percentage of demand the newsvendor seeks to satisfy.

### Searching for an Object

A searcher seeks to find an object in a region divided into an $M$-cell grid. A prior distribution on the object's location initiates the search. At each decision epoch the searcher decides which cell to search. Search is not perfect: if the searched cell contains the object, the searcher finds the item with probability $p_f > 0$ and receives a reward of $R$. Upon finding the object, the searcher terminates the search. Model variants include

multiple within-cell search routines with different costs, search costs that depend on the cell, moving objects, and a maximum search budget.

The POMDP formulation below describes particular model features. It includes some of the same features as the optimal stopping problem in Section 2.8 and may be classified as an *episodic* model.

**Decision Epochs:** Since the object is not found with certainty, the number of decision epochs to find it is unbounded. Therefore

$$T = \{1, 2, \dots\}$$

If search were perfect, at most $M$ epochs would be required.

**Hidden States:** States represent the cell number that contains the object and a stopped state $\Delta$ corresponding to the search being completed after finding the object.

$$S = \{1, 2, \dots, M, \Delta\}.$$

**Actions:** Actions represent the cell to search at each decision epoch. Once the object is found the decision maker may only choose the zero-cost action $\phi$ leaving it in the stopped state. So

$$A_s = \begin{cases} \{1, \dots, M\} & s \neq \Delta \\ \{\phi\} & s = \Delta \end{cases}$$

**Observations:** Set $O = \{F, NF, \Delta_o\}$ where $F$ indicates "object found", $NF$ indicates "object not found" and $\Delta_o$ corresponds to being in the stopped state. The searcher observes $F$ only when successfully searching the cell that contains the object. Observation probabilities follow. When the search is ongoing, that is when $s \neq \Delta$

$$u(o|s,a) = \begin{cases} p_f & o = F,\ a = s \\ 1 - p_f & o = NF,\ a = s \\ 1 & o = NF,\ a \neq s \end{cases}$$

and when the search has stopped, $u(\Delta_o|\Delta, \phi) = 1$.

**Rewards:** It costs one time unit to search a cell. We distinguish rewards in the stopped state from those in the non-stopped state. When $s \neq \Delta$

$$r(s,a,j) = \begin{cases} R & a = s,\ j = \Delta \\ -1 & a = s,\ j = s \\ -1 & a \neq s. \end{cases}$$

The first case corresponds to finding the object, which results in a transition to the stopped state $\Delta$. The second case corresponds to searching the right cell, but not finding it. The third case corresponds to searching the wrong cell. When the search has stopped, $r(\Delta, \phi, \Delta) = 0$.

**Transition Probabilities:** When $s \neq \Delta$

$$p(j|s,a) = \begin{cases} 1 - p_f & a = s,\ j = s \\ p_f & a = s,\ j = \Delta \\ 0 & a = s,\ j \neq s \\ 1 & a \neq s,\ j = a \\ 0 & a \neq s,\ j \neq a \end{cases}$$

and when $s = \Delta$

$$p(j|s,a) = \begin{cases} 1 & a = \phi,\ j = \Delta \\ 0 & a = \phi,\ j \neq \Delta. \end{cases}$$

To obtain transition probabilities we reason as follows. The hidden state $s$ represents the true but unknown location of the object. The system state only changes state when the searcher finds the object, in which case the system moves to the stopped state $\Delta$. This occurs with probability $p_f$ when the state containing the object is searched. No other transitions are possible corresponding to the 0 probability above, If the searcher searches a state that does not contain the object, that is $a \neq s$, the state does not change. Once in the stopped state the system remains there forever.

### Minimally Invasive Surgery

An interesting example of an optimal search problem arises in robotic or minimally invasive surgery. A surgeon seeks to maneuver instruments from an initial incision on the surface of the body to a target organ while avoiding damage to critical structures such as arteries along the way. Pre-operative images provide some indication of the location of these structures but their precise location might change during the surgery. The surgeon has two search options; quick cuts through overlying tissue and slowly peeling away layers of tissue to safely learn what lie behind and a bound on the time to reach the organ.

### Wordle

Bertsekas paper state doesn't change and we're just trying to learn it

another version: state does change

both require changing the state to a probability distribution and we're trying to learn it.

### 3.4.3 Multi-Armed Bandits

Bandit models have received a great deal of attention in the literature; they are simple to describe, widely applicable and have produced some elegant mathematical results. The expression "bandit" refers to a slot machine, which is common place in casinos and gambling halls. A player puts money into the slot machine, pulls its arm, and receives an uncertain payoff that most likely is zero. Playing a particular machine has no effect on other machines the gambler could have chosen to play. Questions of interest are which machine to play, and whether/when to switch to other machines.

Bandit models abstract key features of this process. At each decision epoch, the decision maker selects one of $K$ bandits to play or "arms to pull". We classify bandit models on the basis of how they generate rewards:

1. The *classical statistical bandit*, in which a sample from a fixed probability distribution with unknown mean determines the reward.

2. The *observable Markov chain bandit*, in which the state of an $M$-state Markov chain determines the reward.

3. The *partially observable Markov chain bandit*, in which the decision maker receives both a reward and an observation that depends on the state of the underlying Markov chain.

Some applications assume no cost for switching between arms while others do.

#### Multi-Armed Bandit Applications

Most bandit models explore a trade-off between *exploitation* and *exploration*. The decision maker may either exploit the current machine (continuing pulling its arm) or explore other machines (pull other arms) to find one with a higher payoff. Specific examples include:

1. **Optimal Foraging:** In an attempt to maximize survival, an animal may either continue to forage in the current area or move to another area (patch) with the hope of acquiring more nutrition while expending less energy. The bandit model applies where an "arm" corresponds to which patch to explore. One complication is that changing patches expends energy.

2. **Project Management:** Faced with $K$ projects in progress, at each decision epoch, the decision maker decides which project to work on. As an example, consider the challenge faced by the authors of this book. At each work session they must decide which chapter to work on. Some may proceed quickly, while others may take more background research. By working on one chapter, no progress is made on the others. The reward is the incremental contribution to finishing the selected chapter.

3. **Controlled Clinical Trials:** Randomized controlled clinical trials have become the gold standard for establishing the effective of new drugs, therapies and medical procedures. In a clinical trial patients are randomly assigned to treatments and a control with the goal of determining which treatment is most effective and which have a larger effect than placebo. Randomization controls for potential confounding variables. When conducted sequentially, the challenge is to develop a mechanism for assignment of a patient to a treatment "arm". The use of a multi-arm bandit framework has been shown to efficiently determine the best treatment and as well assign the most patients to the best treatment during the trial.

4. **A/B Testing:** Similarly to a controlled clinical trial, A/B testing refers two a controlled statistical experiment with two "treatments" referred to as Treatment A and Treatment B. While deeply rooted in the statistical literature, it has recently become a widely used approach for choosing between alternative web site designs and online marketing campaigns such as the form and content of banner ads on web pages. Microsoft and Google each conduct over 10,000 A/B test annually. When used sequentially, this may be viewed as a two-armed bandit. In the case of online advertising, when a user clicks a website link, the decision maker chooses whether to display ad design A or ad design B, seeking to determine which design generates the most clicks or the most revenue. Subsequently, new designs are introduced and compared to the previous best design. Multi-armed bandits apply when there are more than two designs to compare.

### POMDP Formulation of two Multi-Armed Bandit Models

Next, we formulate two distinct bandit models: the classical statistical bandit and the partially observable Markov chain bandit. The fully observable Markov chain bandit is a special case of the partially observable one.

### Classical Statistical Bandit Formulation

In the classical statistical bandit model, arm $k$, $k = 1, \ldots, K$ corresponds to a draw from a discrete probability distribution $p(\cdot|\mu_k)$ with mean $\mu_k$ on a discrete set $O = \{o_1, o_2, \ldots, o_L\}$ where $L$ may be finite or infinite. When $O = \{0, 1\}$ the model is referred to as a Bernoulli bandit and applies to settings when the outcome may be viewed as "success' or "failure" such as in A/B testing or clinical trials. We assume for ease of exposition that for $k = 1, \ldots, K$, $\mu_k$ has values in the set finite $M = \{m_1, \ldots, m_J\}$. In a continuous version of this problem, $O$ and/or $M$ may be continuous and probability densities replace probability mass functions.

**Decision Epochs:** The model may be either finite or infinite horizon.

$$T = \{1, 2, \ldots, N\}, \quad N \leq \infty$$

**Hidden States:** Hidden states represent the true but unknown mean for each of the $K$ arms. Thus

$$S = M \times M \times \cdots \times M = M^K = \{(s_1, \ldots, s_K) \mid s_k \in M, k = 1, \ldots, K\}$$

**Actions:** Actions represent which arm to play

$$A = \{1, \ldots, K\}$$

**Observations:** When choosing the $k$-th arm, the decision maker observes the value $o \in O$ with probability $p(o|\mu_k)$ so that

$$u(o|(s_1, \ldots, s_K), k) = p(o|\mu_k)$$

As noted above, in the case of the Bernoulli bandit, $O = \{0, 1\}$ and $\mu_k$ equals the probability of observing a 1.

**Reward:** Assuming no cost for switching arms,

$$r((s_1, \ldots, s_K), k) = f(o)$$

where $f(\cdot)$ is a known real valued function on $O$. Usually we assume $f(o) = o$.

**Transition Probabilities:** Regardless of the action chosen, $\mu_k$ does not change, so that for each $a \in A$

$$p((s_1, s_2, \ldots, s_K)|(s_1, s_2, \ldots, s_K), a) = \begin{cases} 1 & \text{for } s_k \in M \text{ and } k = 1, 2, \ldots, K \\ 0 & \text{otherwise} \end{cases}$$

Implementation of this model requires an initial belief state or prior distribution on $\mu_k$ for each arm. When the decision maker plays arm $k$, only that distribution is updated.

**(apr 1. this still feels pretty abstract. maybe we can make a numerical example)**

**Partially Observable Markov Chain Bandit Formulation**

The state of arm $k$, $k = 1, \ldots, K$ changes according to a Markov chain with state space $S_k$ and transition probability $p_k(j|s)$. We assume the following timing of events. After selecting arm $k$ the decision maker receives a a reward $r_k(s)$ and an observation $o$ with probability $u_k(o|s)$ if the hidden state of arm $k$ equals $s$. Subsequently the arm changes state to state $j$ with probability $p_k(j|s)$.

Note that in order to ensure that the decision maker receives no information from the reward, we can assume each $r_k(\cdot)$ has the same range or that $r_k(\cdot) = r(\cdot)$ for $k = 1, \ldots, K$. Note that this model may be viewed as choosing between $K$ independent unobservable Markov reward processes.

**Decision Epochs:** The model may be either finite or infinite horizon.

$$T = \{1, 2, \ldots, N\}, \quad N \leq \infty$$

**Hidden States:**

$$S = S_1 \times S_2 \ldots \times S_K = \{(s_1, \ldots, s_K) \mid s_k \in S_k, k = 1, \ldots, K\}$$

**Actions:** Actions represent which arm to play so that

$$A = \{1, \ldots, K\}.$$

**Observations:** When the decision maker chooses action $k \in A$ and Markov chain $k$ is in state $s$ the decision maker observes $o \in O$ with probability $u_k(o|s)$. Therefore

$$u(o|(s_1, \ldots, s_K), k) = u_k(o|s_k).$$

This means that the observation depends on the state of arm $k$.

**Rewards:** When the decision maker chooses action $k \in A$ and Markov chain $k$ is in state $s$, the decision maker receives reward

$$r((s_1, \ldots, s_K), k) = r_k(s_k).$$

**Transition Probabilities:** For $k \in A$,

$$p((s'_1, \ldots, s'_K)|(s_1, \ldots, s_K), k) = \begin{cases} p_k(s'_k|s_k) & \text{for } s'_i = s_i, i \neq k \\ 0 & \text{otherwise} \end{cases}$$

When the state space for each Markov Chain is $M$-dimensional, the POMDP model has $M^K$ states, so the model might appear intractable. We show below that the model decomposes into $K$ independent problems that can be analyzed independently. **(where do we show this?)**

### 3.4.4 Breast Cancer Screening

More than one in eight women will be diagnosed with breast cancer in the their lifetime. Using mammography to screen for breast cancer has been shown to reduce breast cancer mortality through early detection. Fundamental issues are at what age to start screening and how frequently to screen. Since woman have different risk factors for breast cancer, screening policies may differ between individuals of the same age. The particular model we develop below is an example of one in which some of the transition probabilities depend on the observation.

Consider the situation where a woman's true health state, that is, whether or not she has breast cancer and possibly what type, is unknown. Mammography provides a noisy preliminary indication of breast cancer. The model recommends periodically whether a woman should or should not have a mammogram. If the no mammogram option is chosen, the patient is encouraged to use self examination as a rough indicator of the presence of cancer.

Action choice involves trade-offs. The downsides of too frequent mammograms are that they are painful, expose a patient to radiation, and false positives may heighten the patient's anxiety and unnecessarily tax health system resources. On the other hand, not having them frequently enough may miss the presence of disease. The goal is to develop a personalized screening policy so that diagnosis and subsequent timely treatment occurs as early as possible. The model is limited to screening. It assumes that after a positive mammogram the patient undergoes a biopsy, which is a more invasive and a highly reliable test. A positive biopsy results in the start of treatment; a negative biopsy returns the patient to their screening regimen. The model can be generalized to include follow-up decisions and treatment policies.

We consider the following sequence of events.

1. Immediately prior to the current decision epoch, the decision maker updates the belief state.

2. The decision maker chooses either mammography or self-examination in the current period.

3. If the decision maker chooses a mammogram it is found to be positive or negative.

4. A positive mammogram results in a biopsy. If the biopsy is also positive, treatment starts prior to the next period. A negative biopsy indicates the patient does not have breast cancer.

5. If the decision maker chose self-examination, the result is revealed prior to the next decision epoch.

6. The health state (cancer free, cancer present, or death) is updated.

**Decision Epochs:** Decisions are made relative to a woman's age. Assume screening starts at age $T_0$ and decisions are made periodically up to age $T_F$. Hence

$$T = \{T_0, T_1, \ldots, T_F\}.$$

Ayer et al. choose $T_0$ to represent age 40, $T_1$ to represent age 40.5 and continue twice-yearly up to age $T_F = 100$. The reality that a woman may die before age 100 is accounted for by adding an absorbing "death" state to the hidden state space.

**Hidden States:** States represent a woman's health condition at each decision epoch. The set of hidden states is given by

$$S = \{CF, BC, TR, DE\}$$

where $CF$ represents cancer free, $BC$ represents breast cancer present, $TR$ represents a patient under treatment and $DE$ represents death. Assuming treatment begins once cancer is diagnosed, states $TR$ and $DE$ are observable and absorbing, while $CF$ and $BC$ are unobservable. Cancer states can be further distinguished by the stage of breast cancer, or whether the cancer is localized versus invasive, if needed.

**Actions:** In hidden states $CF$ and $BC$, the decision maker chooses from $MA$ (mammography) and $SE$ (self examination). In observable states $TR$ and $DE$, the decision maker may only choose an artificial action $\phi$, which leaves the process in the same state. Therefore:

$$A_s = \begin{cases} \{SE, MA\} & \text{for } s \in \{CF, BC\} \\ \{\phi\} & \text{for } s \in \{TR, DE\} \end{cases}$$

**Observations:** Set $O = \{+, -, \Delta\}$ where $+$ denotes a positive mammogram or self-exam, $-$ denotes a negative test and $\Delta$ denotes the screening process has terminated. The probabilities of these observations are given for $t = T_0, T_1, \ldots, T_F$ by

$$u_t(o|s,a) = \begin{cases} \alpha_{t,SE} & \text{for } o = -, s = CF, a = SE \\ 1 - \alpha_{t,SE} & \text{for } o = +, s = CF, a = SE \\ 1 - \beta_{t,SE} & \text{for } o = -, s = BC, a = SE \\ \beta_{t,SE} & \text{for } o = +, s = BC, a = SE \\ \alpha_{t,MA} & \text{for } o = -, s = CF, a = MA \\ 1 - \alpha_{t,MA} & \text{for } o = +, s = CF, a = MA \\ \beta_{t,MA} & \text{for } o = +, s = BC, a = MA \\ 1 - \beta_{t,MA} & \text{for } o = -, s = BC, a = MA \\ 1 & \text{for } o = \Delta, s \in \{TR, DE\}, a = \phi \\ 0 & \text{otherwise.} \end{cases}$$

Observation probabilities are based on the medical testing concepts of *specificity* and *sensitivity*. In this case, the specificity of a test is the probability of a negative test $(\alpha_{t,SE}, \alpha_{t,MA})$ in a cancer-free woman while the sensitivity is the probability of a positive test $(\beta_{t,SE}, \beta_{t,MA})$ in a woman with cancer. These probabilities are obtainable from the medical literature. Moreover, the sensitivity and specificity appear to be age related so that the observation probabilities vary with decision epoch. In general, for mammography to be a useful screening test, its sensitivity and specificity must exceed that of self examination.

**Rewards:** The units for rewards are *Quality Adjusted Life Years* (QALYs). We allow the reward to depend on the observation. Then

$$r_t(s,a,o) = \begin{cases} L_t & \text{for } s \in \{CF, BC\}, a = SE, o \in \{+,-\} \\ L_t - k & \text{for } s \in \{CF, BC\}, a = MA, o = - \\ L_t - K & \text{for } s = CF, a = MA, o = + \\ q_t & \text{for } s = BC, a = MA, o = + \\ 0 & \text{for } s = \{TR, DE\}, a = \phi, 0 = \Delta \end{cases}$$

with a terminal reward $r_{T_F}(s) = q_{T_F}(s)$ for $s \in S$. In the above, $L_t$ denotes the expected QALYs in the period following decision epoch $t$, $k$ the disutility of having a mammogram, $K$ the disutility of a positive mammogram, and $q^t$ the estimated QALYs for an age $t$ woman beginning cancer treatment (which implicitly includes the disutility of a positive mammogram).

**Transition Probabilities:** For $t = T_0, T_1, \ldots, T_F$,

$$p_t(s'|s,a,o) = \begin{cases} (1-\lambda_t)\delta_t & \text{for } s' = BC, s = CF, a = SE, o \in \{+,-\} \\ (1-\lambda_t)(1-\delta_t) & \text{for } s' = CF, s = CF, a = SE, o \in \{+,-\} \\ 1 & \text{for } s' = TR, s = BC, a = MA, o = + \\ (1-\lambda_t) & \text{for } s' = BC, s = BC, a = MA, o = - \\ (1-\lambda_t)\delta_t & \text{for } s' = BC, s = CF, a = MA, o = + \\ (1-\lambda_t)\delta_t & \text{for } s' = CF, s = CF, a = SE, o = + \\ \lambda_t & \text{for } s' = DE, s \in \{CF, BC\}, a = SE, o \in \{+,-\} \\ 1 & \text{for } s' = s, s = \{TR, DE\}, a = \phi, o = \Delta \\ 0 & \text{otherwise} \end{cases}$$

where $\lambda_t$ denotes the probability a woman dies in the period following decision epoch $t$ and $\delta_t$ denotes the probability an age $t$ cancer-free woman develops cancer in the period following decision epoch t. A transition from $BC$ to $TR$ can only occur if a woman with cancer has a positive mammogram.

To apply this model requires obtaining patient-specific data from the medical literature, modifying belief state updating and optimality equations to account for **(rewards and transition probabilities that depend on the observation (is this what you wanted to say?))**, and developing a computational algorithm. A policy will partition the first two components of the belief state (probability of being cancer free and probability of breast cancer) into regions where the action choice is mammography versus self examination. Presumably, the model will recommend a mammogram when the risk of cancer is high and self examination when the probability is low.

### 3.4.5 Controlling Autonomous Robots with Noisy Sensors

An autonomous controlled robot (ACR) consists of sensors to detect features of its environment, a controller to use this information to determine appropriate actions, and an actuator to implement the action. Sensors may use either beams of light or sound to acquire information about its environment. In addition, the robot may have a low-level object avoidance system that works independently of the controller.

In the idealized model we describe, the ACR strives to complete a task of moving through a series of hallways or a maze to reach a target location. We assume the robot has a stored map of its environment and knows the location of the target. Randomness enters through imprecise movements and noisy sensor readings. In robotic navigation competitions, it has been noted that robots often get confused as to their location and consequently are unable to attain their targets. Thus, intelligent control is essential.

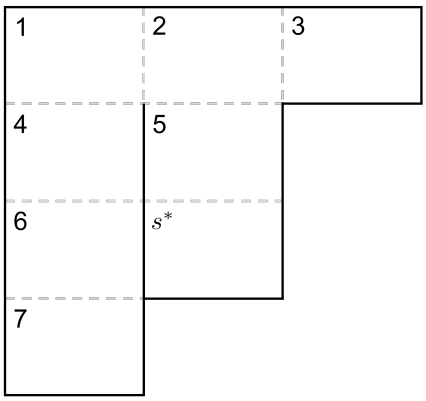


Figure 3.5: Sample grid for robot navigation POMDP model. Thick lines indicate walls.

POMDP models provided a framework for analyzing such problems. The following combines features of many models in the literature. The environment (see Figure 3.5 for an example) is divided into discrete cells, which may be enclosed by up to three walls. This might be regarded as movement through the corridors of an office environment or through a maze. Given a map of the area to be navigated, the ACR can either move or activate its sensors. An onboard compass orients the robot. Since the robot does not know its precise location it uses sensor readings to update its belief distribution.

**Decision Epochs:** Decision epochs correspond to the instance after the robot analyzes its sensor information, if sought. The number of periods to achieve the target is not known a priori. Good policies should achieve the target in finitely many periods, but cycling is possible so an infinite horizon is required to formulate the model.

$$T = \{1, 2, \dots\}$$

**Hidden States:** In an environment with $K$ states plus a target state $s^*$,

$$S = \{1, \ldots, K, s^*\},$$

where the state denotes the robot's actual location.

**Actions:** Actions model the robot's movement and sensing capabilities. For example we may set

$$A = \{U, D, L, R, UD, LR\}$$

where $U$, $D$, $L$, $R$ denote the robot trying to move one cell north, south, west and east, respectively. The robot can only "try" to move in those directions because its movement may be hindered by the presence of a wall. The impact of this impediment will be described in the transition probabilities. Actions $UD$ and $LR$ refer to the robot looking for walls in the north-south and east-west direction, respectively.

**Observations:** The robot's sensors provide noisy information about the number of walls in the direction searched. We include two additional observations, $\phi$ and $\gamma$, which denote no observation and having reached its target, respectively. So, $O = \{0, 1, 2, \phi, \gamma\}$.

We denote by $w_{i,j}$ the probability the robot interprets the sensor information as there being $j$ walls when in fact there are $i$ walls. For example, when there are two walls in the directions searched, $w_{2,j}$ may be described by

$$w_{2,j} = \begin{cases} 0.05 & j = 0, \\ 0.15 & j = 1, \\ 0.8 & j = 2. \end{cases}$$

For sensors to be effective, we would expect $w_{i,i}$ for $i = 0, 1, 2$ to be the largest probabilities. When a move action is designated, no sensor information is provided, so $\phi$ is observed. Thus,

$$u(o|s, a) = \begin{cases} 1 & a \in \{U, D, L, R\} \text{ and } o = \phi \\ 1 & s = s^* \text{ and } o = \gamma \\ w_{W_a(s),j} & s \neq s^*, a \in \{UD, LR\}, j \in \{0, 1, 2\} \end{cases}$$

where $W_a(s)$ denotes the number of walls in the specified direction in cell $s$.

**Rewards:** Several reward structures are possible. They do not depend on the action choice, only the outcome of the move. Most generally

$$r(s, a, j) = \begin{cases} R & s \neq s^*, j = s^* \\ -c & s \neq s^*, j \neq s^* \\ 0 & s = s^*, j = s^*, \end{cases}$$

where $R$ denotes the reward for reaching the target state and $c$ denotes the cost per move. Assuming the process terminates when the target state is reached, the robot remains there and receives no reward.

**Transition Probabilities:** Transition probabilities depend on the precise orientation and configuration of cells in the grid. When sensing, the robot doesn't move so that

$$p(j|s,a) = \begin{cases} 1 & j = s, a \in \{NS, EW\} \\ 0 & j \neq s, a \in \{NS, EW\} \end{cases}$$

Assume that the robot's movement has some randomness to it. When it tries to move in a direction it moves in that direction with probability 0.7 and moves in each other direction with probability 0.1. If it bumps into a wall it does not move. For the environment in Figure 3.5, some selected probabilities are given below:

$$p(j|6,U) = \begin{cases} 0.7 & j = 4 \\ 0.1 & j = 7 \\ 0.2 & j = 6 \end{cases}, \quad p(j|2,D) = \begin{cases} 0.7 & j = 5 \\ 0.1 & j \in \{1,2,3\} \end{cases}, \quad p(4|4,R) = 1$$

For all $a \in A$

$$p(j|s^*,a) = \begin{cases} 1 & j = s^* \\ 0 & j \neq s^* \end{cases}$$

The equivalent MDP of this POMDP can be formulated with actions that are actions in the underlying MDP. That is, we can assume that every action is possible in every state, $\tilde{A} = A$ for all $s \in S$, since infeasible actions (e.g., asking the robot to walk through a wall) lead to the robot to stay in the same cell. This logic is encoded in the transition probabilities as described above.

As an alternative, consider the situation where a robot is not able to detect an infeasible action before implementing it, and thus would attempt to walk through a wall if asked to do so, which is undesirable. In this case, the equivalent MDP needs to be formulated with actions that are decision rules in the underlying MDP, $\tilde{A} \subset \Pi_{s \in S} A_s$ and $A_s$ includes only feasible actions in hidden state $s$.

## 3.5 Finite Horizon Model

In this section we formulate the finite horizon optimization model for a POMDP. We provide Bellman equations and a backwards algorithm algorithm to find optimal values and policies. We also explore the structure of the optimal value function. We illustrate these results with examples. Our development uses results from Chapter **??**. We assume $S$ and $A_s$ for each $s \in S$ are finite (and discrete) and stationary to simplify exposition.

### 3.5.1 Optimality Criterion

Recall that in the derived MDP formulation of the POMDP model, action choice occurs after the decision maker updates the belief state so that a decision rule is a function of the belief state that takes values in the action set. As a result of Theorem **??**, we only need to consider Markovian deterministic policies to obtain optimal solutions, so we restrict attention to such policies to simplify exposition.

Denote the expected total reward of Markovian deterministic policy $\pi = (d_1, d_2, \ldots, d_{N-1})$ given an initial belief state $\mathbf{b} \in \tilde{S}$ by

$$v^{\pi}(\mathbf{b}) := E^{\pi}\left[\sum_{n=1}^{N-1} \tilde{r}(\mathbf{b}_n, d_n(\mathbf{b}_n)) + \tilde{r}_N(\mathbf{b}_N) \;\middle|\; \mathbf{b}_1 = \mathbf{b}\right], \tag{3.44}$$

where $\tilde{r}_N(\mathbf{b}_N)$ is defined by (3.20). Furthermore, denote the expected total reward of policy $\pi$ from decision epoch $n$ to the end of the planning horizon by

$$v_n^{\pi}(\mathbf{b}) := E^{\pi}\left[\sum_{i=n}^{N-1} \tilde{r}(\mathbf{b}_i, d_i(\mathbf{b}_i)) + \tilde{r}_N(\mathbf{b}_N) \;\middle|\; \mathbf{b}_n = \mathbf{b}\right]. \tag{3.45}$$

**Definition 3.2.** An *optimal policy* $\pi^* \in \Pi^{\mathrm{HR}}$ satisfies

$$v^{\pi^*}(\mathbf{b}) \geq v^{\pi}(\mathbf{b}) \tag{3.46}$$

for all $\pi \in \Pi^{\mathrm{HR}}$ and $\mathbf{b} \in \tilde{S}$.

**Definition 3.3.** The *value* of the POMDP is defined by

$$v^*(\mathbf{b}) := \sup_{\pi \in \Pi^{\mathrm{HR}}} v^{\pi}(\mathbf{b}). \tag{3.47}$$

for all $\mathbf{b} \in \tilde{S}$. Similarly, we define the optimal value from epoch $n$ to the end of the planning horizon as

$$v_n^*(\mathbf{b}) := \sup_{\pi \in \Pi^{\mathrm{HR}}} v_n^{\pi}(\mathbf{b}). \tag{3.48}$$

The next result shows that the supremum is attained by a Markovian deterministic policy.

**Theorem 3.2.** There exists $\pi^* \in \Pi^{\mathrm{MD}}$ that achieves the optimal value $v^*(\mathbf{b})$ for

all $\mathbf{b} \in \tilde{S}$. That is,

$$v^*(\mathbf{b}) = \max_{\pi \in \Pi^{\mathrm{MD}}} v^{\pi}(\mathbf{b}). \tag{3.49}$$

**(May 27 Since v is linear in b and $\tilde{S}$ is compact, max is attained. Need to formalize this.)**

### 3.5.2 Computing Optimal Values and Finding Optimal Policies

Next, we present a recursion that allows us to compute $v^*(\mathbf{b}), \mathbf{b} \in \tilde{S}$ in principle. However, since $\tilde{S}$ is a continuum, direct implementation is a challenge. We return to this issue later.

Consider the derived MDP formulation of the POMDP model given in Section 3.3. Given a belief state $\mathbf{b}$, choosing an action $\mathbf{a}$ and observing $o$, the process transitions to a new belief state $\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})$ with probability $\eta(o|\mathbf{b}, \mathbf{a})$ and accrues reward $\tilde{r}(\mathbf{b}, \mathbf{a})$. Thus, given any policy $\pi = (d_1, d_2, \ldots, d_{N-1}) \in \Pi^{\mathrm{MD}}$, we can write $v_n^{\pi}(\mathbf{b})$ as

$$v_n^{\pi}(\mathbf{b}) = \tilde{r}(\mathbf{b}, d_n(\mathbf{b})) + \sum_{o \in O} \eta(o|\mathbf{b}, d_n(\mathbf{b})) v_{n+1}^{\pi}(\boldsymbol{\tau}(o, d_n(\mathbf{b}), \mathbf{b})). \tag{3.50}$$

Thus, the finite horizon Bellman equation for the derived MDP can be written as

$$v_n^*(\mathbf{b}) = \max_{\mathbf{a} \in \tilde{A}} \left\{ \tilde{r}(\mathbf{b}, \mathbf{a}) + \sum_{o \in O} \eta(o|\mathbf{b}, \mathbf{a}) v_{n+1}^*(\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})) \right\}. \tag{3.51}$$

Let $d_n^*(\mathbf{b})$ achieve the maximum in equation (3.51). Then $\pi^* = (d_1^*, \ldots, d_{N-1}^*)$ is an optimal policy with value $v^{\pi^*}(\mathbf{b}) = v^*(\mathbf{b})$, for each $\mathbf{b} \in \tilde{S}$. The following algorithm pro-

vides a high-level perspective on the implementation of value iteration for a POMDP.

**Algorithm 3.1:** The POMDP Finite Horizon Policy Optimization Algorithm

Set $n = N$ and $v_N^*(\mathbf{b}) = \tilde{r}_N(\mathbf{b})$ for all $\mathbf{b} \in \tilde{S}$.
**while** $n > 1$ **do**
  $n \leftarrow n - 1$
  **for** $\mathbf{b} \in \tilde{S}$ **do**
    Evaluate $v_n^*(\mathbf{b})$ according to

$$v_n^*(\mathbf{b}) = \max_{\mathbf{a} \in \tilde{A}} \left\{ \tilde{r}(\mathbf{b}, \mathbf{a}) + \sum_{o \in O} \eta(o|\mathbf{b}, \mathbf{a}) v_{n+1}^*(\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})) \right\}. \tag{3.52}$$

    Set

$$\tilde{A}_{n,\mathbf{b}}^* = \arg\max_{\mathbf{a} \in \tilde{A}} \left\{ \tilde{r}(\mathbf{b}, \mathbf{a}) + \sum_{o \in O} \eta(o|\mathbf{b}, \mathbf{a}) v_{n+1}^*(\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})) \right\}. \tag{3.53}$$

    Select $d_n(\mathbf{b}) \in \tilde{A}_{n,\mathbf{b}}^*$.
**return** $v_1^*(\mathbf{b})$ for all $\mathbf{b} \in \tilde{S}$ and $\pi = (d_1, \ldots, d_{N-1})$.

Some comments about this algorithm follow:

1. The main challenge in implementing this algorithm is that the right hand side of (3.52) must be evaluated for a non-finite number of belief states $\mathbf{b}$. We address this in several ways below.

2. Observe that the summation in (3.52) is indexed by the observation set $O$. This is because transition probabilities in this model, as given by (3.23), only assume only a finite number of positive values indexed again by elements of $O$.

### Structure of POMDP Value Functions

We now investigate properties of $v_n^*(\mathbf{b})$ that facilitate computation for finite horizon models. We begin with a definition.

**Definition 3.4.** Let $\mathbf{x} \in X$ be an $n$-dimensional vector. A real function $f(\mathbf{x})$ on $X$ is a *piecewise linear convex (PLC) function* if for some finite set, $\Gamma$, of $n$-dimensional vectors

$$f(\mathbf{x}) = \max_{\boldsymbol{\gamma} \in \Gamma} \{\boldsymbol{\gamma}^\mathsf{T} \mathbf{x}\}. \tag{3.54}$$

The above definition states that piecewise linear convex functions can be written

as the maximum of a finite number of linear functions[6]. Useful properties of piecewise linear convex functions follow, which we leave as exercises to prove:

1. If $f(\mathbf{x})$ and $g(\mathbf{x})$ are PLC, then $f(\mathbf{x}) + g(\mathbf{x})$ is PLC.
2. Let $\boldsymbol{\gamma}_{\mathbf{x}} \in \arg\max_{\boldsymbol{\gamma}\in\Gamma}\{\boldsymbol{\gamma}^{\mathsf{T}}\mathbf{x}\}$. If $f(\mathbf{x})$ is PLC, then $f(\mathbf{x}) = (\boldsymbol{\gamma}_{\mathbf{x}})^{\mathsf{T}}\mathbf{x}$ for each $\mathbf{x} \in X$.
3. For each $\boldsymbol{\gamma} \in \Gamma$, $\{\mathbf{x} \in X | f(\mathbf{x}) = \boldsymbol{\gamma}^{\mathsf{T}}\mathbf{x}\}$ is convex or empty.

The following theorem establishes an important result on the structure of the optimal value function.

**Theorem 3.3.** Suppose $v_n^*(\mathbf{b})$ is as defined in Algorithm 3.1. Let $\mathbf{r}_N$, $\mathbf{r}_{\mathbf{a}}$, $\mathbf{P}_{\mathbf{a}}$, $\mathbf{U}_{\mathbf{a}}^o$, and $\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})$ be as defined in Section 3.3.1.

1. For $n = 1, \ldots, N$
$$v_n^*(\mathbf{b}) = \max_{\boldsymbol{\gamma}\in\Gamma_n}\{\boldsymbol{\gamma}^{\mathsf{T}}\mathbf{b}\}, \tag{3.55}$$
where $\Gamma_N = \{\mathbf{r}_N\}$,
$$\Gamma_n = \left\{ \mathbf{r}_{\mathbf{a}} + \sum_{o\in O} \mathbf{P}_{\mathbf{a}}\mathbf{U}_{\mathbf{a}}^o \boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})} \;\middle|\; \mathbf{a} \in \tilde{A} \right\} \tag{3.56}$$
for $n = 1, \ldots, N-1$, and
$$\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})} \in \arg\max_{\boldsymbol{\gamma}\in\Gamma_{n+1}}\{\boldsymbol{\gamma}^{\mathsf{T}}\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})\} \tag{3.57}$$
Consequently, $v_n^*(\mathbf{b})$ is a piecewise linear convex function of $\mathbf{b}$.
2. For each $n = 1, \ldots, N$, the state space $\tilde{S}$ can be partitioned into at most $|\Gamma_n|$ polyhedra[a]. Furthermore, all states in a polyhedron have the same optimal action.

[a] Recall that a polyhedron is a set defined by the intersection of a finite number of linear inequalities.

*Proof.* The proof is by (backward) induction on $n$. For $n = N$, $v_N(\mathbf{b}) = \mathbf{r}_N^{\mathsf{T}}\mathbf{b}$, which is a linear function of $\mathbf{b}$, so it is PLC.

[6] In $\Re^1$, an equivalent definition is that $f$ is continuous, convex and can be written as
$$f(x) = \begin{cases} c_1 x + d_1, & x \in [a_1, b_1) \\ c_2 x + d_2, & x \in [a_2, b_2) \\ \cdots & \\ c_m x + d_m, & x \in [a_m, b_m). \end{cases}$$

Assume now that $v_{n+1}(\mathbf{b}), n = 1, \ldots, N-1$ is PLC. This means that for each $\mathbf{b}' \in \tilde{S}, v_{n+1}(\mathbf{b}') = \max_{\boldsymbol{\gamma}\in\Gamma} \boldsymbol{\gamma}^{\mathsf{T}}\mathbf{b}'$ for some finite set $\Gamma$. From (3.23), for each $\mathbf{b} \in \tilde{S}$ and $\mathbf{a} \in \tilde{A}$ at time $n$, there are only finitely many values for the subsequent belief state $\mathbf{b}' = \boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})$ at time $n+1$, each corresponding to a different $o \in O$.

So

$$v_n^*(\mathbf{b}) = \max_{\mathbf{a}\in\tilde{A}} \left\{ \tilde{r}(\mathbf{b}, \mathbf{a}) + \sum_{o\in O} \eta(o|\mathbf{b}, \mathbf{a}) v_{n+1}^*(\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})) \right\} \tag{3.58}$$

$$= \max_{\mathbf{a}\in\tilde{A}} \left\{ \tilde{r}(\mathbf{b}, \mathbf{a}) + \sum_{o\in O} \eta(o|\mathbf{b}, \mathbf{a}) \left( \max_{\boldsymbol{\gamma}\in\Gamma} \boldsymbol{\gamma}^{\mathsf{T}} \boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b}) \right) \right\} \tag{3.59}$$

$$= \max_{\mathbf{a}\in\tilde{A}} \left\{ \mathbf{r}_{\mathbf{a}}^{\mathsf{T}}\mathbf{b} + \sum_{o\in O} \eta(o|\mathbf{b}, \mathbf{a}) \left( \max_{\boldsymbol{\gamma}\in\Gamma} \boldsymbol{\gamma}^{\mathsf{T}} \left( \frac{\mathbf{U}_{\mathbf{a}}^o \mathbf{P}_{\mathbf{a}}^{\mathsf{T}} \mathbf{b}}{\eta(o|\mathbf{b}, \mathbf{a})} \right) \right) \right\} \tag{3.60}$$

$$= \max_{\mathbf{a}\in\tilde{A}} \left\{ \mathbf{r}_{\mathbf{a}}^{\mathsf{T}}\mathbf{b} + \sum_{o\in O} \left( \max_{\boldsymbol{\gamma}\in\Gamma} \boldsymbol{\gamma}^{\mathsf{T}} \left( \mathbf{U}_{\mathbf{a}}^o \mathbf{P}_{\mathbf{a}}^{\mathsf{T}} \mathbf{b} \right) \right) \right\}. \tag{3.61}$$

The expression inside the braces is the summation of a linear function of $\mathbf{b}$ with the sum (over $o$) of a set of piecewise linear convex functions, which results in a piecewise linear convex function. The max over $\mathbf{a}$ preserves piecewise linear convexity, so $v_n^*(\mathbf{b})$ is a piecewise linear convex function of $\mathbf{b}$.

To derive the form of $\Gamma_n$, define $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})} \in \arg\max_{\boldsymbol{\gamma}\in\Gamma_{n+1}} \{\boldsymbol{\gamma}^{\mathsf{T}} \boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})\}$. Then we can write $v_n^*(\mathbf{b})$ as

$$v_n^*(\mathbf{b}) = \max_{\mathbf{a}\in\tilde{A}} \left\{ \left( \mathbf{r}_{\mathbf{a}} + \sum_{o\in O} \mathbf{P}_{\mathbf{a}} \mathbf{U}_{\mathbf{a}}^o \boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})} \right)^{\mathsf{T}} \mathbf{b} \right\}. \tag{3.62}$$

To prove the second result, let us write the finite set of vectors in $\Gamma_n$ as $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K$. For an arbitrary $\boldsymbol{\gamma}_i$ in this set, consider all $\mathbf{b}$ such that $\mathbf{v}_n^*(\mathbf{b}) = \boldsymbol{\gamma}_i'\mathbf{b}$. This set can be written

$$\{\mathbf{b} \mid \boldsymbol{\gamma}_i'\mathbf{b} \geq \boldsymbol{\gamma}'\mathbf{b}, \ \forall \boldsymbol{\gamma} \in \Gamma_n\} \tag{3.63}$$

which is a polyhedron since it is the intersection of a finite number of halfspaces. Since $\boldsymbol{\gamma}_i$ corresponds to a particular action, that action is optimal for all states in this polyhedron.

□

Some remarks about this result:

1. Although the state space is infinite dimensional, we have a finite dimensional characterization of the value function, where it is defined as the maximum of a finite set of linear functions on the unit simplex in $\Re^{|S|}$. Note that this result holds when we have a finite observation set and a finite action set.

2. While each $\boldsymbol{\gamma} \in \Gamma_n$ is associated with a single action, the same action may be associated with multiple $\boldsymbol{\gamma}$, as we will see in Example 3.1.

3. This result is an example of the development in Section **??**, where we discuss optimality of structured policies. Here, we use the special structure of $v^*n(\mathbf{b})$ being piecewise linear convex to show that an optimal policy is piecewise constant over the same pieces that define $v^*n(\mathbf{b})$.

Figure 3.3 illustrates the ideas in this theorem in a simple one-period model ($N = 2$) with two states. The value function is piece-wise linear and the one-dimensional belief space (the probability of being in the first state) is separated into two intervals, one in which the first action is optimal and one in which the second action is optimal. Note that where these intervals meet, at $q = 11/17$, both actions are optimal. A more complex example is provided next.

### Computation

As noted above, the non-finiteness of $\tilde{S}$ makes computation challenging. Most exact algorithms are based on Theorem 3.3. We present a transparent one here and refer to references for more detail and extensions. We also describe some approximations that are easy to follow and might give precise answers.

**An exact algorithm** Theorem 3.3 shows that $v_n^*(\mathbf{b})$ can be defined as the maximum over a finite set of linear functions $\max_{\boldsymbol{\gamma}\in\Gamma_n}\{\boldsymbol{\gamma}^\mathsf{T}\mathbf{b}\}$. We can construct $\Gamma_n$ for each $n$ assuming that we have already constructed $\Gamma_{n+1}$, and initialized with $\Gamma_N = \{\mathbf{r}_N\}$. However, the computations quickly become complex given that $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})}$, which is required to compute $\Gamma_n$, depends on $o, \mathbf{a}$, and $\mathbf{b}$. In other words, to compute $\Gamma_n$ exactly, we would need to know which elements of $\Gamma_{n+1}$ constitute the argmax in equation (3.57) for each value of $o, \mathbf{a}$, and $\mathbf{b}$.

Rather than generating $\Gamma_n$ exactly using $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})}$, the idea is to generate a set $\bar{\Gamma}_n \supseteq \Gamma_n$, and then prune the unnecessary elements. Generating $\Gamma_n$ exactly may require further pruning anyway, since some of the elements may not define $v_n^*(\mathbf{b})$ for any $\mathbf{b}$. Thus, if we can generate $\bar{\Gamma}_n$ efficiently and prune efficiently, we can avoid determining $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})}$ for each value of $o, \mathbf{a}$, and $\mathbf{b}$.

Let $\bar{\Gamma}_n$ be defined as

$$\bar{\Gamma}_n := \left\{ \mathbf{r}_\mathbf{a} + \sum_{o\in O} \mathbf{P}_\mathbf{a}\mathbf{U}_\mathbf{a}^o\bar{\boldsymbol{\gamma}}_o \;\middle|\; \mathbf{a} \in \tilde{A}, \bar{\boldsymbol{\gamma}}_o \in \bar{\Gamma}_{n+1} \right\}. \tag{3.64}$$

with the same starting condition that $\bar{\Gamma}_N = \{\mathbf{r}_N\}$. The subtle difference is that we simply use all elements of $\bar{\Gamma}_{n+1}$, to define $\bar{\Gamma}_n$, even if they would not have been the appropriate element for a particular $o, \mathbf{a}$, and $\mathbf{b}$.

Finally, we must prune the elements of $\bar{\Gamma}_n$ that do not achieve the maximum in $\max_{\bar{\boldsymbol{\gamma}} \in \bar{\Gamma}_n} \boldsymbol{\gamma}^\mathsf{T} \mathbf{b}$ for any $\mathbf{b}$. This can be done by solving the following linear program for every $\bar{\boldsymbol{\gamma}} \in \bar{\Gamma}_n$:

$$\begin{aligned} \text{minimize} \quad & z - \bar{\boldsymbol{\gamma}}^\mathsf{T} \mathbf{b} \\ \text{subject to} \quad & z \geq \boldsymbol{\gamma}^\mathsf{T} \mathbf{b}, \quad \forall \boldsymbol{\gamma} \in \bar{\Gamma}_n \\ & \mathbf{b} \in \tilde{S}. \end{aligned} \tag{3.65}$$

The objective function of this linear program is bounded below by zero, given the first set of constraints. The second set of constraints enforces that $\mathbf{b}$ be in the unit simplex. At optimality, one of the constraints $z \geq \boldsymbol{\gamma}^\mathsf{T} \mathbf{b}$ must be tight. Thus, for a given $\bar{\boldsymbol{\gamma}}$, if the optimal objective value, $z^*$, is strictly greater than zero, then for every $\mathbf{b}$, there exists a $\boldsymbol{\gamma} \in \bar{\Gamma}_n$ such that $z^* = \boldsymbol{\gamma}^\mathsf{T} \mathbf{b} > \bar{\boldsymbol{\gamma}}^\mathsf{T} \mathbf{b}$.

**(PAUSED HERE. Next step is to write out the alg formally. then adjust the long example to reflec this alg. )**

In the example above, it should be clear that $\Gamma_n$ as defined in (3.56) may include more elements than needed to define $v_n^*$. Hence, an algorithm that only generates the necessary elements in $\Gamma_n$ may improve computational efficiency.

**Example 3.1.** We use the two-state example from Section 3.3.2 with $N = 3$ to illustrate Theorem 3.3. To make the example more illuminating, we make a slight change in the reward function, where $r(s_2, a_{2,1}, s_2)$ is changed from $-5$ to 4. With this change, $r(s_2, a_{2,1}) = 4$ and $\mathbf{r}_\mathbf{a}$ is the column vector $(3, 4)$ for $\mathbf{a} = (a_{1,1}, a_{2,1})$.

For $n = 3$, $\Gamma_3 = \{\mathbf{r}_3\} = \{(0, 0)\}$ by definition, since the terminal costs are zero. Then, $v_3^*(\mathbf{b}) = \max_{\boldsymbol{\gamma} \in \Gamma_3} \boldsymbol{\gamma}^\mathsf{T} \mathbf{b} = 0$.

For $n = 2$, we first compute $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})}$ using equation (3.57). Since $\Gamma_{n+1} = \{\mathbf{0}\}$, $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})} = \mathbf{0}$ for all $o, \mathbf{a}$, and $\mathbf{b}$. Next, we compute $\Gamma_2$. Since $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})} = \mathbf{0}$, using equation (3.56), $\Gamma_2$ simplifies to $\{\mathbf{r}_{(a_{1,1},a_{2,1})}, \mathbf{r}_{(a_{1,2},a_{2,2})}\} = \{(3, 4), (5, 2)\}$. The value function is

$$v_2^*(\mathbf{b}) = \max_{\boldsymbol{\gamma} \in \Gamma_2} \boldsymbol{\gamma}^\mathsf{T} \mathbf{b} = \max\{3q_1 + 4q_2, 5q_1 + 2q_2\}.$$

The first term in the max corresponds to action $(a_{1,1}, a_{2,1})$ and the second term corresponds to action $(a_{1,2}, a_{2,2})$. Using the equivalent representation $q_1 = q$ and $q_2 = 1 - q$, we can write the value function as

$$v_2^*(\mathbf{b}) = \max\{-q + 4, 3q + 2\},$$

which is easily visualized in Figure 3.6. Thus, it is clear that

$$\arg\max_{\boldsymbol{\gamma} \in \Gamma_2}{}^{\mathsf{T}} \mathbf{b} = \begin{cases} (a_{1,1}, a_{2,1}), & q \leq 0.5 \\ (a_{2,1}, a_{2,2}), & q > 0.5. \end{cases}$$

That is, for belief states where the probability of being in state $s_1$ is less than 0.5, action $(a_{1,1}, a_{2,1})$ should be chosen, otherwise, choose action $(a_{2,1}, a_{2,2})$.

For $n = 1$, we again compute $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})}$ using equation (3.57). Since $\Gamma_2 = \{(3,4),(5,2)\}$, $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})}$ will now depend on $o$, $\mathbf{a}$, and $\mathbf{b}$.

Consider $o_1$ and $(a_{1,1}, a_{2,1})$. We previously computed that

$$\boldsymbol{\tau}(o_1, (a_{1,1}, a_{2,1}), (q_1, q_2)) = \left( \frac{0.64q_1}{0.72q_1 + 0.4q_2}, \frac{0.08q_1 + 0.4q_2}{0.72q_1 + 0.4q_2} \right)$$

For convenience, let us define

$$q_1' = \frac{0.64q_1}{0.72q_1 + 0.4q_2},$$

and $q_2' = 1 - q_1'$, so we can write

$$\boldsymbol{\tau}(o_1, (a_{1,1}, a_{2,1}), (q_1, q_2)) = (q_1', 1 - q_1').$$

Then

$$\arg\max_{\boldsymbol{\gamma} \in \Gamma_2} \boldsymbol{\gamma}^{\mathsf{T}} \boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b}) = \arg\max\{3q_1' + 4q_2', 5q_1' + 2q_2'\}$$

is determined by whether $q_1'$ is less than or greater than 0.5. Setting $q_1' = 0.5$ is equivalent to $q_1 = 5/12$. In other words, $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o_1,(a_{1,1},a_{2,1}),(q_1,q_2))}$ equals $(3,4)$ if $q_1 \leq 5/12$ and $(5,2)$ if $q_1 > 5/12$. At $q_1 = 5/12$, both vectors satisfy the arg max and we arbitrarily choose $(3,4)$. Proceeding similarly for the other combinations of $o$ and $\mathbf{a}$, we can summarize $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})}$ as follows:

$$\hat{\boldsymbol{\gamma}}_1 := \boldsymbol{\gamma}_{\boldsymbol{\tau}(o_1,(a_{1,1},a_{2,1}),(q_1,q_2))} = \begin{cases} (3,4) & q_1 \leq 5/12 \\ (5,2) & q_1 > 5/12 \end{cases}$$

$$\hat{\boldsymbol{\gamma}}_2 := \boldsymbol{\gamma}_{\boldsymbol{\tau}(o_2,(a_{1,1},a_{2,1}),(q_1,q_2))} = \begin{cases} (3,4) & q_1 \leq 5/16 \\ (5,2) & q_1 > 5/16 \end{cases}$$

$$\hat{\boldsymbol{\gamma}}_3 := \boldsymbol{\gamma}_{\boldsymbol{\tau}(o_1,(a_{1,2},a_{2,2}),(q_1,q_2))} = \begin{cases} (3,4) & q_1 \leq 1/6 \\ (5,2) & q_1 > 1/6 \end{cases}$$

$$\hat{\boldsymbol{\gamma}}_4 := \boldsymbol{\gamma}_{\boldsymbol{\tau}(o_2,(a_{1,2},a_{2,2}),(q_1,q_2))} = (3,4) \text{ for all } q_1 \in [0,1]$$

Note that we arrive at $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o_2,(a_{1,2},a_{2,2}),(q_1,q_2))} = (3,4)$ for all $q_1$ because $\tau(o_2, (a_{1,2}, a_{2,2}), (q_1, q_2))(s_1)$ is less than 0.5 for all values of $q_1 \in [0,1]$.

By definition,

$$\Gamma_1 = \{\mathbf{r}_{(a_{1,1},a_{2,1})} + \mathbf{P}_{(a_{1,1},a_{2,1})} \mathbf{U}^{o_1}_{(a_{1,1},a_{2,1})} \hat{\boldsymbol{\gamma}}_1 + \mathbf{P}_{(a_{1,1},a_{2,1})} \mathbf{U}^{o_2}_{(a_{1,1},a_{2,1})} \hat{\boldsymbol{\gamma}}_2,$$
$$\mathbf{r}_{(a_{1,2},a_{2,2})} + \mathbf{P}_{(a_{1,2},a_{2,2})} \mathbf{U}^{o_1}_{(a_{1,2},a_{2,2})} \hat{\boldsymbol{\gamma}}_3 + \mathbf{P}_{(a_{1,2},a_{2,2})} \mathbf{U}^{o_2}_{(a_{1,2},a_{2,2})} \hat{\boldsymbol{\gamma}}_4\}.$$

To generate all the elements of $\Gamma_1$, we must consider the ranges of $q_1$ where $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},(q_1,q_2))}$ is (3,4) versus (5,2). When $q_1 \leq 5/12$, $\hat{\boldsymbol{\gamma}}_1 = \hat{\boldsymbol{\gamma}}_2 = (3,4)$. When $q_1 \in (5/12, 15/16]$, $\hat{\boldsymbol{\gamma}}_1 = (5,2)$ and $\hat{\boldsymbol{\gamma}}_2 = (3,4)$. Finally, when $q_1 \in (15/16, 1]$, $\hat{\boldsymbol{\gamma}}_1 = \hat{\boldsymbol{\gamma}}_2 = (5,2)$. Hence, action $(a_{1,1}, a_{2,1})$ contributes three vectors to $\Gamma_1$. We illustrate the computation of the second vector. If $q_1 \in (5/12, 5/16]$, then

$$\mathbf{r}_{(a_{1,1},a_{2,1})} + \mathbf{P}_{(a_{1,1},a_{2,1})}\mathbf{U}^{o_1}_{(a_{1,1},a_{2,1})}\hat{\boldsymbol{\gamma}}_1 + \mathbf{P}_{(a_{1,1},a_{2,1})}\mathbf{U}^{o_2}_{(a_{1,1},a_{2,1})}\hat{\boldsymbol{\gamma}}_2 =$$
$$\begin{bmatrix}3\\4\end{bmatrix} + \begin{bmatrix}0.8 & 0.2\\0 & 1\end{bmatrix}\begin{bmatrix}0.8 & 0\\0 & 0.4\end{bmatrix}\begin{bmatrix}5\\2\end{bmatrix} + \begin{bmatrix}0.8 & 0.2\\0 & 1\end{bmatrix}\begin{bmatrix}0.2 & 0\\0 & 0.6\end{bmatrix}\begin{bmatrix}3\\4\end{bmatrix} = \begin{bmatrix}7.32\\7.2\end{bmatrix}$$

For the second action $(a_{1,2}, a_{2,2})$, when $q \leq 1/6$, then $\hat{\boldsymbol{\gamma}}_3 = (5,2)$ and $\hat{\boldsymbol{\gamma}}_4 = (3,4)$, and when $q > 1/6$, $\hat{\boldsymbol{\gamma}}_3 = \hat{\boldsymbol{\gamma}}_4 = (3,4)$. Thus, the second action contributes two vectors to $\Gamma_1$.

Putting it all together, $\Gamma_1$ can be written as:

$$\Gamma_1 = \left\{\begin{bmatrix}6.2\\8\end{bmatrix}, \begin{bmatrix}7.32\\7.2\end{bmatrix}, \begin{bmatrix}7.4\\6\end{bmatrix}, \begin{bmatrix}8.2\\5.76\end{bmatrix}, \begin{bmatrix}9\\5.6\end{bmatrix}\right\},$$

where the first three vectors correspond to action $(a_{1,1}, a_{2,1})$ and the last two correspond to action $(a_{1,2}, a_{2,2})$.

Finally,

$$\begin{aligned} v_1^*(\mathbf{b}) &= \max_{\boldsymbol{\gamma}\in\Gamma_1} \boldsymbol{\gamma}^\mathsf{T}\mathbf{b} \\ &= \max\{6.2q_1 + 8q_2, 7.32q_1 + 7.2q_2, 7.4q_1 + 6q_2, 8.2q_1 + 5.76q_2, 9q_1 + 5.6q_2\} \\ &= \max\{-1.8q + 8, 0.12q + 7.2, 1.4q + 6, 2.44q + 5.76, 3.4q + 5.6\}, \end{aligned}$$

where we have written $q_1 = q$ and $q_2 = 1 - q_1$. The value function can be easily visualized in Figure 3.7. Note that the value function is piecewise linear convex with three pieces, the pieces defined by the vectors (6.2,8), (7.32, 7.2), and (9, 5.6). The first breakpoint occurs at $q = 5/12$ and the second breakpoint occurs at $q = 20/41$. On both sides of the first breakpoint, the optimal action is $(a_{1,1}, a_{2,1})$, so the fact that a breakpoint exists there is due to the fact that $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o_1,(a_{1,1},a_{2,1}),(q_1,q_2))}$ changes from (3,4) to (5,2), which can be observed in its definition above. The second breakpoint is due to a change in the optimal action, where for $q < 20/41$, $(a_{1,1}, a_{2,1})$ is optimal, and for $q > 20/41$, $(a_{1,2}, a_{2,2})$ is optimal.

**One-pass algorithm.** Theorem 3.3 shows that $v_n^*(\mathbf{b})$ can be defined as the maximum over a finite set of linear functions $\max_{\gamma\in\Gamma_n}\{\gamma^\mathsf{T}\mathbf{b}\}$. This first algorithm constructs a sufficient $\Gamma_n$ for each $n$, assuming that we have already constructed $\Gamma_{n+1}$, and initialized with $\Gamma_N = \{\mathbf{r}_N\}$. In the example above, it should be clear that $\Gamma_n$ as defined in (3.56) may include more elements than needed to define $v_n^*$. Hence, an algorithm that only generates the necessary elements in $\Gamma_n$ may improve computational efficiency.

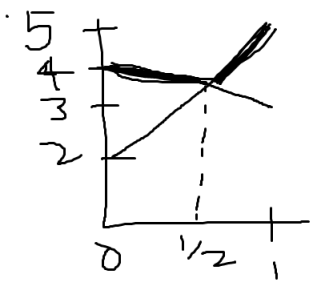


Figure 3.6: Value function for $n = 2$.

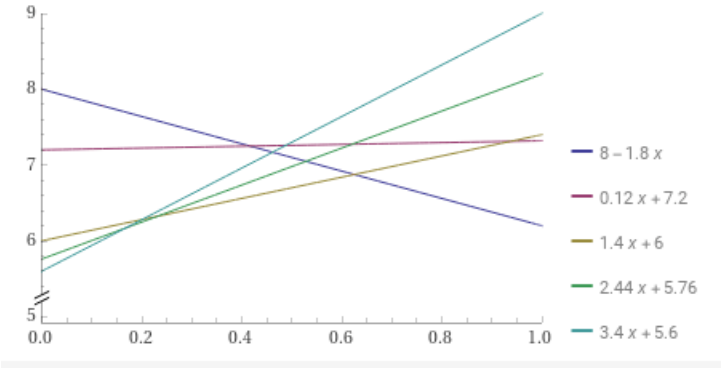


Figure 3.7: Value function for $n = 1$. Redraw and change labels.

For convenience, let us define

$$\boldsymbol{\alpha}_{\mathbf{a}}(\mathbf{b}) := \mathbf{r}_{\mathbf{a}} + \sum_{o \in O} \mathbf{P}_{\mathbf{a}} \mathbf{U}^{o}_{\mathbf{a}} \boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b})}, \tag{3.66}$$

so that we may write $v^*_n(\mathbf{b}) = \max_{\mathbf{a} \in \tilde{A}} \boldsymbol{\alpha}_{\mathbf{a}}(\mathbf{b})^{\mathsf{T}} \mathbf{b}$. Consider an arbitrary $\mathbf{b}_0 \in \tilde{S}$. Let $\boldsymbol{\alpha}^*(\mathbf{b}_0)$ be such that $\boldsymbol{\alpha}^*(\mathbf{b}_0)^{\mathsf{T}} \mathbf{b}_0 = \max_{\mathbf{a} \in \tilde{A}} \boldsymbol{\alpha}_{\mathbf{a}}(\mathbf{b}_0)^{\mathsf{T}} \mathbf{b}_0$ and define $\mathbf{a}^* \in \arg\max_{\mathbf{a} \in \tilde{A}} \boldsymbol{\alpha}_{\mathbf{a}}(\mathbf{b}_0)^{\mathsf{T}} \mathbf{b}_0$ to be a corresponding optimal action. Since $v^*_n(\mathbf{b}_0) = \boldsymbol{\alpha}^*(\mathbf{b}_0)^{\mathsf{T}} \mathbf{b}_0$, we can include $\boldsymbol{\alpha}^*(\mathbf{b}_0)$ in $\Gamma_n$.

Let us now define a region around $\mathbf{b}_0$ in which $\boldsymbol{\alpha}^*(\mathbf{b}_0)$ is guaranteed to continue representing the value function. There are two ways the $\boldsymbol{\alpha}$ vector changes. First, the state could move to one in which $\mathbf{a}^*$ is no longer optimal, i.e., where $\boldsymbol{\alpha}^*(\mathbf{b}_0)$ is no longer the vector that achieves the maximization in the definition of the value function. So,

$v_n(\mathbf{b}) = \boldsymbol{\alpha}^*(\mathbf{b}_0)^\mathsf{T}\mathbf{b}$ holds if $\boldsymbol{\alpha}^*(\mathbf{b}_0)^\mathsf{T}\mathbf{b} \geq \boldsymbol{\alpha}_\mathbf{a}(\mathbf{b}_0)^\mathsf{T}\mathbf{b}$ for all $\mathbf{a} \in \tilde{A}$.

Second, even if the optimal action is maintained, the vector $\boldsymbol{\alpha}_\mathbf{a}(\mathbf{b})$ may deviate from $\boldsymbol{\alpha}^*(\mathbf{b}_0)$ if the $\mathbf{b}$ is such that $\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b}) \neq \boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b}_0)$. That is, as we move away from $\mathbf{b}_0$, we may hit a different linear segment in the piecewise linear representation of $v_{n+1}(\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b}))$. At $\mathbf{b}_0$,

$$v_{n+1}(\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b}_0)) = \max_{\boldsymbol{\gamma} \in \Gamma_{n+1}} \boldsymbol{\gamma}^\mathsf{T}\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b}_0) = \boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b}_0)}^\mathsf{T}\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b}_0).$$

As we move away from $\mathbf{b}_0$ to an arbitrary $\mathbf{b}$, for $v_{n+1}(\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b}))$ to remain on the linear segment defined by $\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b}_0)}$, it suffices that

$$\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b}_0)}^\mathsf{T}\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b}) \geq \boldsymbol{\gamma}_i^\mathsf{T}\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b}),$$

for all $\boldsymbol{\gamma}_i \in \Gamma_{n+1}, o \in O, \mathbf{a} \in \tilde{A}$. By definition of $\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})$, this inequality is equivalent to

$$\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b}_0)}^\mathsf{T}\left(\frac{\mathbf{U}_\mathbf{a}^o\mathbf{P}_\mathbf{a}^\mathsf{T}\mathbf{b}}{\eta(o|\mathbf{b}, \mathbf{a})}\right) \geq \boldsymbol{\gamma}_i^\mathsf{T}\left(\frac{\mathbf{U}_\mathbf{a}^o\mathbf{P}_\mathbf{a}^\mathsf{T}\mathbf{b}}{\eta(o|\mathbf{b}, \mathbf{a})}\right)$$

or

$$\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b}_0)}^\mathsf{T}\mathbf{U}_\mathbf{a}^o\mathbf{P}_\mathbf{a}^\mathsf{T}\mathbf{b} \geq \boldsymbol{\gamma}_i^\mathsf{T}\mathbf{U}_\mathbf{a}^o\mathbf{P}_\mathbf{a}^\mathsf{T}\mathbf{b},$$

since $\eta(o|\mathbf{b}, \mathbf{a}) > 0$.

Putting this all together, the following finite set of inequalities define a region around $\mathbf{b}_0$ such that $v_n(\mathbf{b}) = \boldsymbol{\alpha}^*(\mathbf{b}_0)^\mathsf{T}\mathbf{b}$ for all $\mathbf{b}$ in this region:

$$\boldsymbol{\alpha}^*(\mathbf{b}_0)^\mathsf{T}\mathbf{b} \geq \boldsymbol{\alpha}_\mathbf{a}(\mathbf{b}_0)^\mathsf{T}\mathbf{b}, \quad \forall \mathbf{a} \in \tilde{A}, \tag{3.67}$$

$$\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b}_0)}^\mathsf{T}\mathbf{U}_\mathbf{a}^o\mathbf{P}_\mathbf{a}^\mathsf{T}\mathbf{b} \geq \boldsymbol{\gamma}_i^\mathsf{T}\mathbf{U}_\mathbf{a}^o\mathbf{P}_\mathbf{a}^\mathsf{T}\mathbf{b}, \quad \forall \boldsymbol{\gamma}_i \in \Gamma_{n+1}, o \in O, \mathbf{a} \in \tilde{A} \tag{3.68}$$

$$\mathbf{b}^\mathsf{T}\mathbf{e} = 1, \tag{3.69}$$

$$\mathbf{b} \geq \mathbf{0}. \tag{3.70}$$

Note that some of these constraints may be superfluous in that they are not needed to define the region. Let (3.67) and (3.68) be represented compactly as the following $K$ inequalities

$$\mathbf{b}^\mathsf{T}\mathbf{g}^k \geq 0, \quad k = 1, \ldots, K. \tag{3.71}$$

Superfluous constraints can be detected by solving the following linear program

$$\begin{aligned} \text{minimize} \quad & \mathbf{b}^\mathsf{T}\mathbf{g}^m \\ \text{subject to} \quad & \mathbf{b}^\mathsf{T}\mathbf{g}^k \geq 0, \quad k = 1, \ldots, K \\ & \mathbf{b}^\mathsf{T}\mathbf{e} = 1 \\ & \mathbf{b} \geq \mathbf{0}. \end{aligned} \tag{3.72}$$

for each $m = 1, \ldots, K$ and noting the value of the slack variable associated with the $k$-th constraint. The slack variable is zero if and only if the constraint is needed to define the polyhedron. If it is positive, the constraint can be removed.

Each of the remaining, binding constraints can be associated with a new $\mathbf{b}$ vector on the boundary of the region, which will identify potentially several $\boldsymbol{\alpha}$ vectors that achieve $\arg\min_{\mathbf{a}\in\tilde{A}} \boldsymbol{\alpha}_{\mathbf{a}}(\mathbf{b})^{\mathsf{T}}\mathbf{b}$. These new $\boldsymbol{\alpha}$ vectors can be added to $\Gamma_n$ and investigated following the above approach. Any previously investigated $\boldsymbol{\alpha}$ vector can be ignored. Eventually, this process will exhaust the entire belief space $\tilde{\mathcal{S}}$, at which point $\Gamma_n$ is complete. Then, the entire process repeats to determine $\Gamma_{n-1}$.

**Sondik's algorithm**

1. **Initialization:** Set $\Gamma_N = \{\mathbf{r}_N\}$, $v_N^*(\mathbf{b}) = \mathbf{r}_N^\mathsf{T}\mathbf{b}, n = N-1$ and $\Gamma_n = \emptyset$. Choose arbitrary $\mathbf{b}_0 \in \tilde{\mathcal{S}}$ and let $T = \{\mathbf{b}_0\}$.

2. **Identify new vector and region:** Determine

$$\mathbf{a}^* \in \arg\max_{\mathbf{a}\in\tilde{A}} \boldsymbol{\alpha}_\mathbf{a}(\mathbf{b}_0)^\mathsf{T}\mathbf{b}_0,$$

   where $\boldsymbol{\alpha}_\mathbf{a}(\mathbf{b}_0)$ is defined as in (3.66).

   $T \leftarrow T \setminus \mathbf{b}_0$.

   If $\boldsymbol{\alpha}_{\mathbf{a}^*}(\mathbf{b}_0) \in \Gamma_n$, proceed to step 4. Else, set $\Gamma_n \leftarrow \Gamma_n \cup \boldsymbol{\alpha}_{\mathbf{a}^*}(\mathbf{b}_0)$.

   Let

$$G := \{\mathbf{b} \in \tilde{\mathcal{S}} \mid (\boldsymbol{\alpha}^*(\mathbf{b}_0) - \boldsymbol{\alpha}_\mathbf{a}(\mathbf{b}_0))^\mathsf{T}\mathbf{b} \geq 0, \forall \mathbf{a} \in \tilde{A};$$
$$(\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b}_0)} - \boldsymbol{\gamma}_i)^\mathsf{T}\mathbf{U}_\mathbf{a}^o\mathbf{P}_\mathbf{a}^\mathsf{T}\mathbf{b} \geq 0, \forall \boldsymbol{\gamma}_i \in \Gamma_{n+1}, o \in O, \mathbf{a} \in \tilde{A}\}.$$

3. **Determine boundary of region:**

   For each $\mathbf{a} \in \tilde{A}$, solve

$$\begin{aligned} &\text{minimize} && (\boldsymbol{\alpha}^*(\mathbf{b}_0) - \boldsymbol{\alpha}_\mathbf{a}(\mathbf{b}_0))^\mathsf{T}\mathbf{b} \\ &\text{subject to} && \mathbf{b} \in G \end{aligned} \tag{3.73}$$

   and let $\mathbf{b}_\mathbf{a}^*$ be an optimal solution. If $(\boldsymbol{\alpha}^*(\mathbf{b}_0) - \boldsymbol{\alpha}_\mathbf{a}(\mathbf{b}_0))^\mathsf{T}\mathbf{b}_\mathbf{a}^* = 0$, $T \leftarrow T \cup \mathbf{b}_\mathbf{a}^*$.

   For each $\mathbf{a} \in \tilde{A}, o \in O, \gamma_i \in \Gamma_{n+1}$, solve

$$\begin{aligned} &\text{minimize} && (\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b}_0)} - \boldsymbol{\gamma}_i)^\mathsf{T}\mathbf{U}_\mathbf{a}^o\mathbf{P}_\mathbf{a}^\mathsf{T}\mathbf{b} \\ &\text{subject to} && \mathbf{b} \in G \end{aligned} \tag{3.74}$$

   and let $\mathbf{b}_{\mathbf{a},o,i}^*$ be an optimal solution. If $(\boldsymbol{\gamma}_{\boldsymbol{\tau}(o,\mathbf{a},\mathbf{b}_0)} - \boldsymbol{\gamma}_i)^\mathsf{T}\mathbf{U}_\mathbf{a}^o\mathbf{P}_\mathbf{a}^\mathsf{T}\mathbf{b}_{\mathbf{a},o,i}^* = 0$, $T \leftarrow T \cup \mathbf{b}_{\mathbf{a},o,i}^*$.

4. **Stopping criterion:**

   If $T = \emptyset$ and $n = 1$, stop.

   If $T = \emptyset$ and $n > 1$, then $n \leftarrow n-1$ and $\Gamma_n = \emptyset$. Choose arbitrary $\mathbf{b}_0 \in \tilde{\mathcal{S}}$, let $T = \{\mathbf{b}_0\}$ and return to step 2.

   If $T \neq \emptyset$, let $\mathbf{b}_0$ be one of the elements in $T$ and return to Step 2.

## 3.6 Infinite Horizon Models

In this section, we extend our development of POMDPs to an infinite horizon, focusing particularly on the discounted case.

### 3.6.1 Optimality Criterion

Let $v_\lambda^\pi(\mathbf{b})$ be the expected total discounted reward of a policy $\pi = (d_1, d_2, \ldots)$, given an initial belief state $\mathbf{b} \in \tilde{S}$:

$$v_\lambda^\pi(\mathbf{b}) := E^\pi \left[ \sum_{n=1}^{\infty} \tilde{r}(\mathbf{b}_n, d_n(\mathbf{b}_n)) \;\middle|\; \mathbf{b}_1 = \mathbf{b} \right]. \tag{3.75}$$

**Definition 3.5.** An *optimal policy* $\pi^* \in \Pi^{\mathrm{HR}}$ satisfies

$$v_\lambda^{\pi^*}(\mathbf{b}) \geq v_\lambda^\pi(\mathbf{b}) \tag{3.76}$$

for all $\pi \in \Pi^{\mathrm{HR}}$ and $\mathbf{b} \in \tilde{S}$.

**Definition 3.6.** The *value* of the POMDP is defined by

$$v_\lambda^*(\mathbf{b}) := \sup_{\pi \in \Pi^{\mathrm{HR}}} v_\lambda^\pi(\mathbf{b}). \tag{3.77}$$

for all $\mathbf{b} \in \tilde{S}$.

Similar to the fully observable case, stationary deterministic policies are optimal within the class of all polices for a POMDP.

**Theorem 3.4.** There exists $\pi^* \in \Pi^{\mathrm{SD}}$ that achieves the optimal value $v_\lambda^*(\mathbf{b})$ for all $\mathbf{b} \in \tilde{S}$. That is,

$$v_\lambda^{\pi^*}(\mathbf{b}) = v_\lambda^*(\mathbf{b}). \tag{3.78}$$

Going forward, we restrict our focus to stationary deterministic policies.

### 3.6.2 Computing Optimal Values and Finding Optimal Policies

- something on policy evaluation - bellman equation - contraction mapping, exists unique sol - value iteration alg, building off piecewise linear characterization. forward

index. superscript for iterate to differentiate from epoch - policy iteration - sondiks alg for finitely transient policies

Extending (3.50) to the discounted infinite horizon setting, we can write the value of a stationary deterministic policy $\pi = (d, d, \ldots)$ as

$$v_\lambda^\pi(\mathbf{b}) = \tilde{r}(\mathbf{b}, d(\mathbf{b})) + \lambda \sum_{o \in O} \eta(o|\mathbf{b}, d(\mathbf{b})) v_\lambda^\pi(\boldsymbol{\tau}(o, d(\mathbf{b}), \mathbf{b})). \tag{3.79}$$

Thus, the optimal value function can be written as

$$v_\lambda^*(\mathbf{b}) = \max_{\mathbf{a} \in \tilde{A}} \left\{ \tilde{r}(\mathbf{b}, \mathbf{a}) + \lambda \sum_{o \in O} \eta(o|\mathbf{b}, \mathbf{a}) v_\lambda^*(\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})) \right\}. \tag{3.80}$$

If $d^*(\mathbf{b})$ achieves the maximum in equation (3.80), then $\pi^* = (d^*, d^*, \ldots)$ is a stationary optimal policy with value $v_\lambda^*(\mathbf{b})$, for each $\mathbf{b} \in \tilde{S}$.

Similar to the infinite horizon discounted (fully observable) MDP case, the Bellman operator here is a contraction mapping, which means that the Banach fixed-point theorem (Theorem **??**) applies and there exists a unique solution to (3.80). This leads to an analogous value iteration algorithm for solving POMDPs.

**Value iteration: component notation**

1. **Initialize:** Set $n = 0$, specify $\epsilon > 0$ and $v^0(\mathbf{b})$ for all $\mathbf{b} \in \tilde{S}$.

2. **Iterate:** Compute $v^{n+1}(\mathbf{b})$ for all $\mathbf{b} \in \tilde{S}$:

$$v^{n+1}(\mathbf{b}) = \max_{\mathbf{a} \in \tilde{A}} \left\{ \tilde{r}(\mathbf{b}, \mathbf{a}) + \lambda \sum_{o \in O} \eta(o|\mathbf{b}, \mathbf{a}) v^n(\boldsymbol{\tau}(o, \mathbf{a}, \mathbf{b})) \right\}. \tag{3.81}$$

3. **Apply stopping criterion:** If **(paused here)**

$$\mathrm{sp}(\mathbf{v}^{n+1} - \mathbf{v}^n) \geq \frac{\lambda}{1 - \lambda} \epsilon,$$

proceed to step 4, else $n \leftarrow n + 1$ and return to step 2.

4. **Choose an $\epsilon$-optimal policy:** For all $s \in S$, choose

$$d^\epsilon(s) \in \arg\max_{a \in A_s} \left\{ r(s, a) + \sum_{j \in S} \lambda p(j|s, a) v^{n+1}(j) \right\}$$

and

$$v^\epsilon(s) = v^n(s) + \lambda(1 - \lambda)^{-1}(\underline{L\mathbf{v} - \mathbf{v}}).$$

## 3.7 Bibliographic Remarks

Our development follows that in the operations research literature. We use the models of Sondik, Sondik and Smallwood, and Monahan ... Minimal invasive surgery motivated research by Shechter et al. Book by Krishnanmurty Robotic control Simmons and Koenig

We describe a generic equipment inspection/replacement problem adapted from Monahan **?**.

(Kohavi, Ron; Thomke, Stefan (September 2017). "The Surprising Power of Online Experiments". Harvard Business Review: 74–82) - for 10000 AB experiments annually ref.

Markov chain bandit model following Krishnamurthy **?**.

The POMDP model we describe was developed by Ayer, Alagoz and Stout to develop personalized screening guidelines.

Machine repair and inspection problem adopted from Monahan review paper.

**(need ref for 1 in 8 women will develop breast cancer in lifetime)**

## 3.8 Exercises

1. For the example in Section 3.1.3 calculate the expected total reward when $p > 0.5$ with terminal reward 0 and when $x = 3$ and $y = 4$.

2. For the one-period, two-state model in Section 3.1.3, compute $\mathbf{b}_2$ for the following combinations of decision rule and observation:

   (a) use $d_1$ and observe $o_2$,

   (b) use $d_2$ and observe $o_1$, and

   (c) use $d_2$ and observe $o_2$.

3. Let $p^*$ denote the point at which the value of the decision rules (3.2) and (3.3) with terminal reward 0 are equal. Plot $v_d(p)$ as a function of $p$ for $p \in [0, 1]$ using the decision rule in (3.2) for $p \leq p^*$ and the decision rule in (3.3) for $p > p^*$. What do you observe about the shape of $v_d(p)$?

4. Compute $P(X_2 = s_1 | Z_2 = o_2)$ for the two-state model using the approach in Section 3.1.3.

5. Compute the reward and transition probabilities in the two-state model for the the remaining actions in $\tilde{A}$ in a similar manner to (3.28) and (3.33).

6. Compute and interpret $\tilde{p}(x|(p_1, p_2), (a_{1,1}, a_{2,2}))$ for the two-state model in a similar way to which (3.33) was computed.

7. Compute and interpret transition probabilities and rewards when $p_1 = 0.4$.

8. Reformulate the preventive maintenance problem when transitions between states occur at the beginning of a period.

9. Formulate an $n$-state version of the preventive maintenance problem.

10. Formulate the derived MDP for the newsvendor problem from Section 3.4.2 when $M = 3$.

11. Formulate the derived MDP for the search problem from Section 3.4.2. This is challenging because of the presence of the stopped state.

12. Reformulate the search problem when the searcher finds the object with certainty.

13. Reformulate the search model when the object can change cells between decision epochs. Assume a the object follows a random walk where $p_L$ denotes the probability the object moves from cell $i$ to cell $i-1$ and $p_H$ denotes the probability the object moves from cell $i$ to cell $i+1$.

14. Formulate the two derived MDPs of the robot control problem described at the end of Section 3.4.5.

15. Prove that the sum of piecewise linear convex functions is piecewise linear and convex. Hint: start with a simple numerical example in two dimensions.

16. Compute the value of using the information seeking action defined by (3.35) in a one-period version of the two-state model.

17. Three state model from Smallwood and Sondik.

18. Provide a matrix representation of the preventative maintenance model and the derived MDP formulation.

19. Derive all probabilities for the robot control model in Section 3.4.5.

# Chapter 4

# Appendices

## 4.1 Notation and conventions

### 4.1.1 General math notation

- := for definition.
- $\Re^N$ for $N$-dimensional Euclidean space. **(prefer mathbb for consistency)**
- $\mathbb{Z}^N$ for $N$-dimensional integer lattice. $\mathbb{Z}_+^N$ restricts the lattice to non-negative integers.
- $I_A$ is the indicator of event $A$, meaning that $I_A = 1$ if $A$ is true and 0 if $A$ is false. **(Check we use this consistently)**
- sup for supremum and max for maximum.
  - The supremum of a real-valued function $f(w)$ over a set $W$ refers to the smallest upper bound of $f(w)$ the elements in the set $\{f(w) \mid w \in W\}$. Importantly, the supremum need not be an element of that set.
  - The maximum denotes the largest element of a set and must be contained in that set.
  - As a convention, we write sup when a set is non-finite and reserve max for finite sets. When we prove that suprema are attained, we replace sup by max.
  - inf and min are the analogues to sup and max, respectively, when we refer to lower bounds.
- $u^+ = \max\{u, 0\}$, given a scalar $u$.
- arg max and arg min for the set of maximizers and minimizers, respectively.

  - For a real-valued function $f(w)$ and a set $W$,

$$\arg\max_{w \in W} f(w) := \{w^* \in W \mid f(w^*) \geq f(w) \text{ for all } w \in W\}.$$

  - arg min is defined similarly as arg max, except with respect to minimization.

- $\mathbf{e}$ a vector with all components equal to one.
- $\mathbf{I}$ is the identity matrix.
- $^\mathsf{T}$ denotes transpose.
- $\emptyset$ is the empty set.
- $|A|$ is the number of elements in a set $A$.
- $o(1)$ is a generic expression for a function $f(n)$ on the non-negative integers, that converges to zero as $n\infty$. It is used when we only are concerned about the limiting property of a function and not its particular form. More generally a function is $o(g)$ if $f(n)/g(n) \to 0$.
- $A^B$ denotes the set of all functions from set $B$ to set $A$. **(do we use this?)**
- $A - B$ denotes the set of elements in set $A$ that are not in subset set $B$.

### 4.1.2 Notation specific to MDPs

**(Must be used in two or more chapters)**

- $w_d(a|s)$ is the probability that randomized decision rule $d$ chooses action $a$ in state $s$
- $p^\pi(\cdot|\cdot)$ - conditional probability of an event under policy $\pi$
- $E^\pi[\cdot|\cdot]$ - conditional expectation of a random variable under policy $\pi$
- $d^\infty$ denotes the stationary policy that uses decision rule $d \in D^{\text{MR}}$ at every decision epoch.
- $d_\mathbf{v}$ denotes a $\mathbf{v}$-improving decision rule.
- c-max denotes component-wise maximum. Applied to a vector, each component of the vector is maximized independently of the others, meaning that the arg max is in general different for each component.
- $L$ value iteration operator.
- $L_d$ is the operator that applied decision rule $d$ for one period.

- $B$ is the operator on a value function $\mathbf{v}$ defined as $L\mathbf{v} - \mathbf{v}$.
- $x(s, a)$ are dual variables in the linear programming formulation of an MDP.
- $\mathbf{B}$ is the "design" matrix in a linear approximation (Chapter **??**)
- $\mathbf{\Gamma}$. it the matrix $(\mathbf{B}^\mathsf{T}\mathbf{B})^{-1\mathsf{T}}$. (Chapter **??**)
- $\mathbf{\Pi}$ is the projection matrix $\mathbf{B}(\mathbf{B}^\mathsf{T}\mathbf{B})^{-1\mathsf{T}}$ (Chapter **??**)
- $\underline{\mathbf{v}} = \min_{s\in S} v(s)$.
- $\overline{\mathbf{v}} = \max_{s\in S} v(s)$.
- $\rho(\mathbf{A})$ denotes the largest eigenvalue of the matrix $\mathbf{A}$ in absolute value. It is referred to as the spectral radius of $\mathbf{A}$.

### 4.1.3 Conventions

- Random variables are denoted by capital letters and their values by lower case letters.
- Vectors and matrices are bolded, while their components are nonbold. A vector written as $\mathbf{x} = (x_1, \ldots, x_n)$ should be considered a column vector, unless it is transposed $\mathbf{x}^\mathsf{T}$ explicitly.
- Subscripts and superscripts are defined based on the context.
  - Subscripts denote epochs for value functions, state-action value functions, rewards and transition probabilities.
  - Superscripts denote epochs for states, actions and histories.
  - Subscripts denote different states, actions or histories, with the sets of states, actions or histories, respectively.
  - Superscripts on a value function denote a specific policy ($\pi$) or an optimal policy ($*$)
  - Subscripts on $v$ denote the optimality criterion being used. **(check this. conflicts with first subbullet)**
- Hat denotes approximation (Chapter **??**).
- Square brackets are reserved for expectation, variance or covariance.
- Stationary optimal policy and optimal stationary policy are two different concepts. The former refers to the fact that the optimal policy under consideration is stationary. The latter refers to the best stationary policy among all stationary policies, but it may not be optimal among all policies, including those that are non-stationary.

- separate convention for infinite horizon models? **(come back to this)**
- Bellman operator

1. cartesian product and power set
2. Series, limits and geometric series
3. lim inf and lim sup
4. Mathematical Induction
5. Some useful lemmas
6. $\sum$ over an arbitrary set.
7. Convexity and support

## 4.2 Abbreviations

- MDP: Markov decision process
- POMDP: Partially observed Markov decision process
- HR: History-dependent and randomized **(need this?)**
- MSE: Mean squared error
- RMSE: Root mean squared error (the square root of the MSE)
- TD: Temporal differencing
- PI: Policy iteration
- MPI: Modified policy iteration
- VI: Value Iteration
- LP: Linear Program
- LSVI: Least Squares Value Iteration
- LSPI: Least Squares Policy Iteration

## 4.3 Markov Chains

Markov chain theory underlies Markov decision process models; especially under the average reward and expected total reward criteria. Markov chains have a long history which we won't go into here. We especially like the seminal book Kemeny and Snell [1960] for its transparent approach and innovative applications. It provides the basis for the vanishing discount approach developed by Blackwell [1962] to analyze average reward models. Chapter 4 of Gallagher [2013] also provides an insightful discussion of the use of eigenvalues and eigenvectors to explore limiting properties of powers of transition probability matrices and of the challenges faced when analyzing countable state Markov chains.

Markov chains are now widely applied and provide the basis for Google's PageRank algorithm, speech recognition software and many reinforcement learning applications.

### 4.3.1 What is a finite Markov chain?

Let $S$ denote a finite set of states and let $\mathcal{X} = \{X_n : n = 0, 1, 2, \ldots\}$ denote a sequence of random variables with values in $S = \{s_1, \ldots, s_m\}$[1]. Then $\mathcal{X}$ is a *Markov chain* if

$$P(X_n = s^n | X_{n-1} = s^{n-1}, X_{n-2} = s^{n-2}, \ldots, X_0 = s^0) = P(X_n = s^n | X_{n-1} = s^{n-1}) \tag{4.1}$$

for all $n = 0, 1, \ldots$ and $s^k \in S$ for $k = 0, 1, \ldots$.

Equation (4.1) is known as the *Markov property* which can be stated succinctly as *"the future conditional on the present is independent of the past"*.

We say that $\mathcal{X}$ is *stationary* or *homogeneous* whenever $P(X_n = s^n | X_{n-1} = s^{n-1})$ is independent of $n$. In this case we define for $k = 0, 1, \ldots$, the *(one-step) transition probability*

$$p(j|s) := P(X_k = j | X_{k-1} = s) \tag{4.2}$$

and the *n-step transition probability* by

$$p_n(j|s) := P(X_{n+k} = j | X_k = s). \tag{4.3}$$

We will let $\mathbf{P}$ denote the $|S| \times |S|$ matrix with $(s, j)$th component $p(j|s)$. Using the law of total probabilities $n$-times shows that this matrix provides provides a convenient way of computing the $n$-step transition probabilities. It follows that $(s, j)$th component of the matrix product $\mathbf{P}^n$ equals $p_n(j|s)$. Also, $\mathbf{P}^0 = \mathbf{I}$.

For $s \in S$ define the initial distribution $q_0(s) := P(X_0 = s)$ and the unconditional distribution $q_n(s) = P(X_n = s)$. Then again by the law of total probabilities, the unconditional distribution

$$q_n(s) = \sum_{j \in S} q_0(j) p_n(s|j)$$

[1]Recall that we use subscripts to denote states and superscripts to denote the state visited at a decision epoch.

which in matrix-vector notation can be written as $\mathbf{q}_n = \mathbf{q}_0\mathbf{P}^n$.

Finally, for a real valued function $r(\cdot)$ on $S^2$ we define the conditional expectation

$$E_s[r(X_n)] := E[r(X_n)|X_0 = s] = \sum_{j\in S} r(j)p_n(j|s).$$

which in matrix-vector notation can be written $E_s[r(X_n)] = \mathbf{P}^n\mathbf{r}(s)$. Note that the expressions $\mathbf{P}^n$ and $\mathbf{P}^n\mathbf{r}$ will appear throughout this this book when using matrices and vectors to analyze stationary policies because:

> A stationary policy $d^\infty$ generates a Markov chain with transition probability matrix $\mathbf{P}_d$ and reward vector $\mathbf{r}_d$.

Consequently, each of the models in Chapter 2 provides an application of a Markov chain.

### 4.3.2 Classifying states

The limiting behavior of the Markov chain depends on relationships between its states. We say state $j$ is *accessible* from state $s$, written $s \rightsquigarrow j$, if $p_n(j|s) > 0$ for some $n \geq 0$. Otherwise $j$ is *inaccessible* from $S$. If $s \rightsquigarrow j$ and $j \rightsquigarrow s$ then states $j$ and $s$ are said to *communicate* which we write as $s \leftrightsquigarrow j$.

We say that state $s \in S$ is:

- *recurrent* if the time to return to state $s$ is finite with probability one. This occurs if state $s$ is accessible from all states that are accessible from $s$. That is, if $s \rightsquigarrow j$, then $j \rightsquigarrow s$ [3].
- *absorbing* whenever $p(s|s) = 1$. This means that once the chain visits state $s$, it remains there forever.
- *transient* if the time to return to state $s$ is finite with probability *less than* one. Equivalently if there exists a state $j$ for which $s \rightsquigarrow j$ but $s$ is not accessible from $j$. That is after a transition to $j$, $p_n(s|j) = 0$ for $n = 0, 1, \ldots$.
- *periodic with period* $m$ if the greatest common divisor of $\{n = 0, 1, \ldots | p_n(s|s) > 0\}$ is $m$.
- *aperiodic* if it is periodic with period 1. .

[2] A Markov chain together with a reward function is often referred to as a *Markov reward process*.

[3] In countable state chains, $s$ is *positive recurrent* if the *expected* time to return to it is finite. Otherwise it is said to be *null recurrent*. In a finite Markov chain all recurrent states are positive recurrent.

Note that if $s$ and $j$ communicate, they will be classified in the same way. That is if $s \leftrightsquigarrow j$ and $s$ is recurrent or periodic with period $m$, than $j$ is recurrent or periodic with period $m$. Moreover starting in a recurrent state $s$, then expected number of returns to state $s$ is infinite, while if $s$ is transient, the expected number of returns to $S$ is finite.

**Example 4.1.** As a simple illustration of a periodic Markov chain, consider one with transition probability matrix

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ p & 0 & 1-p \\ 0 & 1 & 0 \end{bmatrix}. \tag{4.4}$$

where $S = \{s_1, s_2, s_3\}$ and $0 < p < 1$. Since

$$\mathbf{P}^2 = \begin{bmatrix} p & 0 & 1-p \\ 0 & 1 & 0 \\ p & 0 & 1-p \end{bmatrix} \tag{4.5}$$

it follows that $p_2(s_i|s_i) > 0$ for $i = 1, 2, 3$. You can verify that for $n \geq 1$, $\mathbf{P}^{2n+1} = \mathbf{P}$ and $\mathbf{P}^{2(n+1)} = \mathbf{P}^2$ so that it follow that each state is periodic with period 2.

Observe that when $p = 0$ or $p = 1$ the chain behaves differently.

We note that periodicity represents an "edge case" that complicates many analyses and necessitates averaging to ensure convergence (see Section 4.3.5).

### 4.3.3 Classes and class structure

Call a subset $C$ of $S$ *closed* if no state outside of $C$ is accessible from any state in $C$. Moreover $C$ is *irreducible* if no proper subset of $C$ is closed. We say that an irreducible closed set $C$ that consists of a single element is *absorbing*.

A Markov chain can be partitioned into closed irreducible subsets of states $C_1, \ldots, C_M$, in which all states in each $C_i$ are recurrent, and a set of transient states $T$. If the Markov chain starts at a state in some $C_k$, it remains in $C_k$ forever, however if it starts in $T$ it eventually leaves it and ends up in some $C_k$. Obviously when $S$ is finite, the Markov chain contains at least one closed class.

Classification depends on the arrangement of 0 entries in $\mathbf{P}$. For example, consider the transition probability matrix with $S = \{s_1, s_2, s_3, s_4, s_5\}$

$$\mathbf{P} = \begin{bmatrix} a & b & 0 & 0 & 0 \\ c & d & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & e & 0 & 0 & f \\ g & 0 & h & u & v \end{bmatrix}$$

where lower case letters denote non-zero probabilities with row sums equal to one. Consequently $C_1 = \{s_1, s_2\}$, $C_2 = \{s_3\}$ and $T = \{s_4, s_5\}$. Moreover $s_3$ is absorbing. Note that starting in state $s_4$, the system can either jump to $C_1$ in one step or remain in $T$ for a few steps and then jump to either $C_1$ or $C_2$. Observe also the zeroes in rows 1-3 of columns 4 and 5.

A matrix partitioned as above is said to be in *canonical form*. Any transition matrix can be converted to the canonical form[4].

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_2 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot & \cdot \\ \mathbf{0} & \mathbf{0} & \dots & & \mathbf{P}_M & \mathbf{0} \\ \mathbf{Q}_1 & \mathbf{Q}_2 & \dots & & \mathbf{Q}_M & \mathbf{Q}_{M+1} \end{bmatrix}$$

where $\mathbf{P}_k$ corresponds transitions between states in $C_k$ and $Q_{M+1}$ to transitions between states in $T$.

### 4.3.4 Chain structure

Many results in this book, especially in Chapters **??** and **??**, depend on the structure of Markov chains corresponding to stationary policies. A Markov chain on $S$ is said to be:

- *regular*[5] if $S$ is a single closed aperiodic class.
- *recurrent* or *ergodic* if it consists of a single closed class. A recurrent chain may be either periodic or aperiodic. When it aperiodic, it is regular.
- *unichain* if it consists of a single closed class and a non-empty set $T$ of transient states, and
- *multi-chain* if it consists of two or more closed classes and possibly some transient states.

The analysis in the text will focus primarily on models in which all stationary policies correspond to Markov chains which are either recurrent or unichain. In the Markov decision process context, we will refer to them as recurrent or unichain models.

[4]The Fox-Landi algorithm does this, see Section A.3 in Puterman [1994]

[5]Kemeny and Snell [1960] refer to a Markov chain as regular if $\mathbf{P}^N$ has all positive entries for some $N$. Clearly that is equivalent to our notion.

### 4.3.5 Limiting behavior

From an Markov decision process perspective, classification of the limiting behavior of $\mathbf{P}$ is extremely important. We say that a sequence of matrices $\mathbf{P}_n$ converge to a matrix $\mathbf{P}^*$ if for each $s$ and $j$ in $S$, its components $p_n(j|s)$ converges to the components of $\mathbf{P}^*$, $p^*(j|s)$. This is sometimes referred to as component-wise convergence[6].

We state the following important result without proof.

**Theorem 4.1.** Let $\mathbf{P}$ be a the transition probability matrix of a Markov chain on a finite state space $S$.

a. Then the limit
$$\mathbf{P}^* := \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{P}^{n-1} \tag{4.6}$$
exists.

b. $\mathbf{P}^*$ is a transition probability matrix and satisfies
$$\mathbf{P}^*\mathbf{P} = \mathbf{P}\mathbf{P}^* = \mathbf{P}^*\mathbf{P}^* = \mathbf{P}^*. \tag{4.7}$$

c. When $\mathbf{P}$ is recurrent,
$$\mathbf{P}^* = \mathbf{e}\mathbf{q}^T \tag{4.8}$$
where $\mathbf{q}$ satisfies $\mathbf{q}^T = \mathbf{q}^T\mathbf{P}$ subject to $\mathbf{q}^T\mathbf{e} = 1$[a]. Moreover each entry of $\mathbf{q}$ is strictly positive.

d. If $s \in T$, then $\mathbf{P}^*(s|j) = 0$ for all $j \in S$.

e. When $\mathbf{P}$ is regular,
$$\mathbf{P}^* = \lim_{N\to\infty} \mathbf{P}^n \tag{4.9}$$

f. When $\mathbf{P}$ is regular, the convergence in (4.9) is eponentially fast. That is for each $j$ and $s$ in $S$, there exists constants $k > 0$ and $0 \le a < 1$ for which
$$|p_n(j|s) - p^*(j|s)| < ka^n \tag{4.10}$$

[a]Recall that $\mathbf{e}$ denotes a column vector of 1's so that $\mathbf{q}^T\mathbf{e}$ is a scalar and $\mathbf{e}\mathbf{q}^T$ is a $|S| \times |S|$ matrix with all rows equal to $\mathbf{q}$.

We comment on the significance of the above result.

1. The representation of $\mathbf{P}^*$ in (4.6) is particularly relevant in the Markov decision process setting in which $\sum_{n=0}^{N-1} \mathbf{P}^n\mathbf{r}$ denotes the expected total reward over $N$ decision epochs so that $\mathbf{P}^*\mathbf{r}$ equals the *limiting average expected reward*.

[6]This is equivalent to convergence in norm when $S$ is finite.

2. As a consequence of (4.9), in a regular chain, $\mathbf{P}^*\mathbf{r}$ can be interpreted as the *steady state* reward. Note that in a periodic chain such as in (4.5), this limit doesn't exist. However the limit in (4.6) always exists. (See the brief example following this list).

3. Part c./@ provides an approach for computing $\mathbf{P}^*$ for a recurrent chain, that is by solving the system of equations

$$q(s) = \sum_{j \in S} q(j)p(s|j)$$

subject to $\sum_{j \in S} q(j) = 1$. In this case

$$\mathbf{P}^* = \begin{bmatrix} q(s_1) & q(s_2) & \dots & q(s_M) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ q(s_1) & q(s_2) & \dots & q(s_M) \end{bmatrix}$$

The vector $\mathbf{q}$ is called the *stationary distribution* of the Markov chain.

4. The approach described in the previous item applies to computing the stationary distribution and the components of $\mathbf{P}^*$ for any closed class.

5. Part d./@ says that that the long run average time spent in a transient state is zero. When the Markov chain is unichain, this means that that for all $s \in S$

$$p^*(j|s) = \begin{cases} q(j) & j \in C \\ 0 & j \in T \end{cases}$$

we denote the entries of the matrix $\mathbf{P}^*$ by $p^*(j|s)$,

6. Puterman [1994] p.593-594 describes an approach for computing $p^*(j|s)$ for $s \in T$ and $j$ in any closed class.

7. Part f. follows from Gallagher [2013]) which shows that for a regular chain the second largest eigenvalue of $\mathbf{P}$ is strictly less than 1 and equals $a$.

**Example 4.1 ctd.** Since $\mathbf{P}$ is not regular, (4.9) does not hold. Hence we can compute the components of $\mathbf{P}^*$ using part c. to obtain:

$$\mathbf{P}^* = \begin{bmatrix} \frac{p}{2} & \frac{1}{2} & \frac{1-p}{2} \\ \frac{p}{2} & \frac{1}{2} & \frac{1-p}{2} \\ \frac{p}{2} & \frac{1}{2} & \frac{1-p}{2} \end{bmatrix}. \tag{4.11}$$

Observe that when $0 < p < 1$, all rows of $\mathbf{P}^*$ are equal since $\mathbf{P}$ consists of a single closed class. Moreover $\mathbf{P}^* = \frac{1}{2}\mathbf{P} + \frac{1}{2}\mathbf{P}^2$ consistent with its representation in part a. and the observation that for $n \geq 1$, $\mathbf{P}^{2n+1} = \mathbf{P}$ and $\mathbf{P}^{2(n+1)} = \mathbf{P}^2$. Note also that when $p = 1$, states $s_1$ and $s_2$ form a recurrent class of period 2 and $s_3$ is transient. Similarly when $p = 0$, states $s_2$ and $s_3$ form a recurrent class of period 2 and $s_1$ is transient.

### 4.3.6 An important lemma

This section provides an important result about convergence of geometric series of matrices that is fundamental for Chapters **??-??**. It holds in considerable generality and can be proved under a range of hypotheses, but the following finite-dimensional version suffices for this book. Note the proof generalizes that used to derive the scalar identity $\sum_{n=0}^{\infty} a^n = (1-a)^{-1}$ when $|a| < 1$. We follow Kemeny and Snell [1960].

**Lemma 4.1.** Let $\mathbf{Q}$ denote an $|S| \times |S|$ real-valued matrix for which $\mathbf{Q}^N \to \mathbf{0}$ as $N \to \infty$. Then the inverse of $\mathbf{I} - \mathbf{Q}$ exists and satisfies

$$\sum_{n=0}^{\infty} \mathbf{Q}^n = (\mathbf{I} - \mathbf{Q})^{-1} \tag{4.12}$$

*Proof.* Let

$$\mathbf{S}_N := \sum_{n=0}^{N} \mathbf{Q}^n$$

Then its easy to see that

$$(\mathbf{I} - \mathbf{Q})\mathbf{S}_{N-1} = \mathbf{I} - \mathbf{Q}^N.$$

Letting $N \to \infty$ and applying the hypothesis $\mathbf{Q}^N \to \mathbf{0}$ yields

$$(\mathbf{I} - \mathbf{Q}) \sum_{n=0}^{\infty} \mathbf{Q}^n = \mathbf{I}$$

from which the result follows. □

Note that sufficient conditions for (4.12) to hold include $||\mathbf{Q}|| < 1$ and the spectral radius[7] of $\mathbf{Q}$, $\sigma(\mathbf{Q}) < 1$. The latter condition applies to matrices with some row sums greater than 1.

The following generalization, the proof of which we leave as an exercise, will be useful when we require results that apply when $\mathbf{P}$ is periodic. The summation below and in (4.6) are referred to as *Cesaro summation*[8]

**Lemma 4.2.** Let $\mathbf{Q}$ denote an $M \times M$ matrix for which $\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \mathbf{Q}^{N-1} \to \mathbf{0}$. Then

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \left[ \sum_{n=1}^{k} \mathbf{Q}^n \right] = (\mathbf{I} - \mathbf{Q})^{-1} \tag{4.13}$$

We apply the first lemma to $\mathbf{Q} = \lambda\mathbf{P}$ in discounted models, $\mathbf{Q} = \mathbf{P} - \mathbf{P}^*$ in the next sub-section and $\mathbf{Q}$ equal to the sub-matrix of $\mathbf{P}$ on its transient states when analyzing transient models (or stochastic shortest path models) under the expected total reward criterion.

### 4.3.7 The deviation matrix

The *deviation matrix*

$$\mathbf{H}_P := \sum_{n=0}^{\infty} (\mathbf{P}^n - \mathbf{P}^*)$$

Is fundamental to the analysis of the average reward model. We now provide a closed form representation:

**Proposition 4.1.** Let $\mathbf{H}_P$ be defined as above. Then if $\mathbf{P}^n \to \mathbf{P}^*$,

$$\mathbf{H}_P = (\mathbf{I} - (\mathbf{P} - \mathbf{P}^*))^{-1} - \mathbf{P}^* = (\mathbf{I} - (\mathbf{P} - \mathbf{P}^*))^{-1}(\mathbf{I} - \mathbf{P}^*). \tag{4.14}$$

*Proof.* Using equalities in (4.7), it is easy to establish directly or by induction that for $n \geq 1$, $\mathbf{P}^n - \mathbf{P}^* = (\mathbf{P} - \mathbf{P}^*)^n$.

[7] Largest eigenvalue in absolute value

[8] The *Cesaro summation* of the series $x_n, n = 0, 1, 2, \ldots$ equals

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N-1} x_n.$$

It often exists when $\lim_{n\to\infty} x_n$ does not. For example the non-convergent series $1, 0, 1, 0, \ldots$ has a *Cesaro limit* equal to $\frac{1}{2}$. In the Markov chain context, we use (component-wise) Cesaro summation to obtain limits in periodic chains.

Hence

$$\mathbf{H}_P = \sum_{n=0}^{\infty}(\mathbf{P}^n - \mathbf{P}^*) = (\mathbf{I} - \mathbf{P}^*) + \sum_{n=1}^{\infty}(\mathbf{P}^n - \mathbf{P}^*)$$
$$= \mathbf{I} + \sum_{n=1}^{\infty}(\mathbf{P} - \mathbf{P}^*)^n - \mathbf{P}^* = \sum_{n=0}^{\infty}(\mathbf{P} - \mathbf{P}^*)^n - \mathbf{P}^*.$$

Since $\mathbf{P}^n \to \mathbf{P}^*$, $(\mathbf{P} - \mathbf{P}^*)^n$ converges to zero so that the result follows from Lemma 4.1. The equivalence in (4.14) follows from applying (4.7)[9]. □

Note that when the Markov chain is periodic or some recurrent class is periodic then the representation for $\mathbf{H}_P$ in (4.14) is still valid but the summation is in the Cesaro-sense. This means that

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} \sum_{i=0}^{k} \left[\mathbf{P}^k - \mathbf{P}^*\right] = (\mathbf{I} - (\mathbf{P} - \mathbf{P}^*))^{-1} - \mathbf{P}^*.$$

Note that in both cases, $\mathbf{H}_P$ has the following useful and easy to prove properties:

$$(\mathbf{I} - \mathbf{P})\mathbf{H}_P = \mathbf{H}_P(\mathbf{I} - \mathbf{P}) = \mathbf{I} - \mathbf{P}^* \tag{4.15}$$
$$\mathbf{H}_P\mathbf{P}^* = \mathbf{P}^*\mathbf{H}_P = \mathbf{0}. \tag{4.16}$$

### 4.3.8 Structure of $\mathbf{P}^*$ and $\mathbf{H}_P$

Our proofs of convergence of average reward value iteration and policy iteration in Chapter **??** exploit the following properties of $\mathbf{P}^*$ and $\mathbf{H}_P$ in unichain Markov chains.

Let $R$ denote the set of recurrent states and $T$ denote the set of transient states of $\mathbf{P}$. Then P can be written as the partitioned matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{RR} & \mathbf{P}_{RT} \\ \mathbf{P}_{TR} & \mathbf{P}_{TT} \end{bmatrix} \tag{4.17}$$

where the sub-matrix $\mathbf{P}_{RR}$ corresponds to transitions between recurrent states, $\mathbf{P}_{TR}$ corresponds to transitions from transient to recurrent states, $\mathbf{P}_{RT}$ corresponds to transitions from recurrent states to transient states and $\mathbf{P}_{TT}$ corresponds to transitions between transient states. Note that all entries of $\mathbf{P}_{RT}$ equal zero since by definition such transitions cannot occur. The dimensions of these sub-matrices are *conformable*, that is they depend on the number of recurrent and transient states. The matrices $\mathbf{I}$ and $\mathbf{0}$ will be conformable with the partition.

For ease of reference we restate parts c. and d. of Theorem 4.1 as follows:

[9] See Appendix A in Puterman [1994] or Blackwell [1962] for two very different proofs of this result.

**Lemma 4.3.** Let $\mathbf{P}$ be unichain. Then

$$\mathbf{P}^* = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \end{bmatrix} \tag{4.18}$$

where $\mathbf{Q}$ denotes an $|S| \times |R|$ matrix with each row equal to $\mathbf{q}$ as defined in part c. of Theorem 4.1 and $\mathbf{0}$ denotes an $|S| \times |T|$ matrix of zeroes.

The following proposition describes an important property of $\mathbf{P}_{TT}$ that provides the basis for analysis of transient models in Chapter **??**.

**Proposition 4.2.** The matrix $\mathbf{I} - \mathbf{P}_{TT}$ is invertible and satisfies

$$(\mathbf{I} - \mathbf{P}_{TT})^{-1} = \sum_{n=0}^{\infty} \mathbf{P}_{TT}^n \tag{4.19}$$

*Proof.* By the definition of transience, for each $s \in T$, a states in $R$ is accessible from $s$ in a finite number of transitions, that is there exists a $j \in R$ for which $p_n(j|s) > 0$ for some $j \in R$. Since there are finitely many states (in $T$), there exists a $k \geq 1$ for which each row sum of $\mathbf{P}_{TT}^k$ is strictly less than one. Hence from Lemma 4.1 the result follows. □

An immediate consequence of (4.19) is that the $(s, j)$th component of the matrix $(\mathbf{I} - \mathbf{P}_{TT})^{-1}$ equals the expected total number of times the chain is in state $j \in T$ starting from $s \in T$[10].

We now express $\mathbf{H}_P$ in partitioned form as

$$\mathbf{H}_P = \begin{bmatrix} \mathbf{H}_{RR} & \mathbf{H}_{RT} \\ \mathbf{H}_{TR} & \mathbf{H}_{TT} \end{bmatrix}. \tag{4.20}$$

The following lemma describes relevant properties of particular sub-matrices of $\mathbf{H}_P$ for unichain models.

**Lemma 4.4.** Let $\mathbf{P}$ be unichain. Then

a. all entries of $\mathbf{H}_{RT}$ equal zero,

b. $\mathbf{H}_{TT} = (\mathbf{I} - \mathbf{P}_{TT})^{-1}$ and

c. $\mathbf{H}_{TT} \geq \mathbf{I}$.

[10] See Chapter III in Kemeny and Snell [1960].

*Proof.* Writing $\mathbf{Z}_P := \mathbf{I} - (\mathbf{P} - \mathbf{P}^*)$ in partitioned form gives

$$\mathbf{Z}_P = \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{W} & \mathbf{I} - \mathbf{P}_{TT} \end{bmatrix}$$

where $\mathbf{I}$ denotes a $|T| \times |T|$ identity matrix and $\mathbf{0}$ denotes an $|R| \times |T|$ matrix of zeroes and $\mathbf{W}$ and $\mathbf{W}$ are specific matrices we don't care about. Standard formulae for inverting a partitioned matrix establish that

$$\mathbf{Z}_P^{-1} = \begin{bmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{Y} & (\mathbf{I} - \mathbf{P}_{TT})^{-1} \end{bmatrix}$$

where we don't require the form of $\mathbf{X}$ and $\mathbf{Y}$. From Lemma 4.3

$$\mathbf{I} - \mathbf{P}^* = \begin{bmatrix} \mathbf{I} - \mathbf{P}^*_{RR} & \mathbf{0}_{RT} \\ -\mathbf{P}^*_{TR} & \mathbf{I}_{TT} \end{bmatrix}$$

where subscripts suggest the sub-matrix dimensions. Since $\mathbf{H}_P = \mathbf{Z}_P^{-1}(\mathbf{I} - \mathbf{P}^*)$, parts a. and b follow. Part c. follows from Proposition 4.2 and the fact that the entries of $\mathbf{P}_{TT}$ are non-negative. □

### 4.3.9 Some examples

**Example 4.2. A two-state Markov chain**

Suppose $S = \{s_1, s_2\}$ and

$$\mathbf{P} = \begin{bmatrix} a & 1-a \\ 1-b & b \end{bmatrix}.$$

We consider several special cases:

1. If $0 < a < 1$ and $0 < b < 1$, the Markov chain is regular and $\mathbf{P}^n \to \mathbf{P}^*$ where

$$\mathbf{P}^* = \begin{bmatrix} \frac{1-b}{2-a-b} & \frac{1-a}{2-a-b} \\ \\ \frac{1-b}{2-a-b} & \frac{1-a}{2-a-b} \end{bmatrix}$$

2. If $a = 1$ and $0 < b < 1$, then the Markov chain is unichain, $s_1$ is absorbing, $\mathbf{P}^n \to \mathbf{P}^*$ and

$$\mathbf{P}^* = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

as noted in Lemma 4.3.

3. Suppose $a = b = 1$. Then the Markov chain is multi-chain and $\mathbf{P}^n = \mathbf{P}^*$ for all $n = 0, 1, \ldots$.

4. Suppose $a = b = 0$, the chain consists of single closed periodic class (with period 2), $\mathbf{P}^n \nrightarrow \mathbf{P}^*$ but from part c. of Theorem 4.1,

$$\mathbf{P}^* = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{P}^n = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

This means that the average time in each state is $\frac{1}{2}$ which is obvious from the structure of $\mathbf{P}$.

We leave it as an exercise to compute $\mathbf{H}_P$ for each of these cases.

**Example 4.3. A multi-state Markov chain**

We analyze a slightly more complicated example with $S = \{s_1, \ldots, s_6\}$ and

$$\mathbf{P} = \begin{bmatrix} a & 1-a & 0 & 0 & 0 & 0 \\ 1-b & b & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ c & d & e & f & g & h \end{bmatrix}.$$

We assume $0 < a < 1$ and $0 < b < 1$. This multi-chain matrix is in canonical form and corresponds to closed classes $C_1 = \{s_1, s_2\}$, $C_2 = \{s_3, s_4, s_5\}$ and transient class $T = \{s_6\}$. Because $C_2$ is periodic, $\mathbf{P}^n \nrightarrow \mathbf{P}^*$ but we can define $\mathbf{P}^*$ using (4.6).

Letting $w = \frac{1-b}{2-a-b}$, we obtain

$$\mathbf{P}^* = \begin{bmatrix} w & 1-w & 0 & 0 & 0 & 0 \\ w & 1-w & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ \\ (\frac{c+d}{1-h})w & (\frac{c+d}{1-h})(1-w) & (\frac{e+f+g}{1-h})(\frac{1}{3}) & (\frac{e+f+g}{1-h})(\frac{1}{3}) & (\frac{e+f+g}{1-h})(\frac{1}{3}) & 0 \end{bmatrix}.$$

A general formula for deriving the last row appears on p.594 of Puterman [1994] however in this example, we can argue that starting in state $s_6$, the chain ends in $C_1$ with probability $(\frac{c+d}{1-h})$ and ends in $C_2$ with probability $(\frac{e+f+g}{1-h})$. Then the probability of ending in a particular state is the probability of ending in a class times the probability of ending in a particular state in that class.

### 4.3.10 Eigenvalues and eigenvectors of a transition matrix*

Eigenvectors and eigenvalues provide insight into the limiting behavior of Markov chains. We use some basic linear algebra results (for example see Strang [2023] and his brilliant online lectures).

The following result key result will shed much insight into the rate at which a Markov chain converges to its limit.

**Theorem 4.2.** Suppose the $m \times m$ matrix $\mathbf{P}$ has $m$ independent eigenvectors. Then

$$\mathbf{P} = \mathbf{W}\Lambda\mathbf{W}^{-1} \tag{4.21}$$

where $\mathbf{W}$ is a matrix with columns equal to the right eigenvectors of $\mathbf{P}$ and $\Lambda$ is a diagonal matrix with entries equal to eigenvalues of $\mathbf{P}$.

Moreover

$$\mathbf{P}^n = \mathbf{W}\Lambda^n\mathbf{W}^{-1}. \tag{4.22}$$

We illustrate this result with an example and discuss the consequences thereafter.

**Example 4.4.** Let

$$\mathbf{P} = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix}.$$

Note that this matrix corresponds to the decision rule $d(s_1) = a_{1,1}$ and $d(s_2) = a_{2,2}$ in Example **??** and $\mathbf{P}$ is regular.

We leave it as an exercise to check that[a]

$$\Lambda = \begin{bmatrix} 1.0 & 0 \\ 0 & 0.4 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 1.0 & 0.2 \\ 1.0 & -0.4 \end{bmatrix} \quad \text{and} \quad \mathbf{W}^{-1} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{5}{3} & -\frac{5}{3} \end{bmatrix}$$

Expanding the right hand side of (4.21) gives

$$\mathbf{P} = 1 \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix} + 0.4 \begin{bmatrix} \frac{1}{3} & -\frac{1}{3} \\ \frac{2}{3} & -\frac{2}{3} \end{bmatrix} \tag{4.23}$$

it follows from (4.22) that for $n \geq 1$

$$\mathbf{P}^n = 1^n \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix} + 0.4^n \begin{bmatrix} \frac{1}{3} & -\frac{1}{3} \\ \frac{2}{3} & -\frac{2}{3} \end{bmatrix}. \tag{4.24}$$

Therefore

$$\mathbf{P}^n \to \mathbf{P}^* = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix} \tag{4.25}$$

and the convergence is at rate $0.4^n$ as noted in Theorem 4.1.

[a]Recall that the sum of of the eigenvalues equals the trace of a matrix and the product of eigenvalues equals its determinant.

Observe that in this example the eigenvalues of $\mathbf{P}$ are 1 and 0.4. In general we have the following important result often referred to as the *Perron-Frobenius theorem*.

**Theorem 4.3.** Let $\mathbf{P}$ be a transition probability matrix. Then

1. 1 is an eigenvalue of $\mathbf{P}$
2. all eigenvalues of $\mathbf{P}$ are less than or equal to 1 in absolute value, and
3. the eigenvector corresponding to eigenvalue 1 has non-negative components.

Observe also that the stationary distribution $(\frac{2}{3}, \frac{1}{3})$ arises naturally in this example as the left eigenvector of $\mathbf{P}$ corresponding to the eigenvalue 1. It corresponds to right eigenvalue of 1 which equals $(1, 1)^T$. Note also that when $\mathbf{P}$ is regular, all eigenvalues other than 1 are strictly less than 1.

We return to the periodic model in Example 4.2 to illustrate a case where the limit in the previous example does not exist and Theorem 4.3 holds with all eigenvalues equal to 1 in absolute value.

**Example 4.5.** Let

$$\mathbf{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Then

$$\Lambda = \begin{bmatrix} 1 & 0 \\ \\ 0 & -1 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 1 & 1 \\ \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{W}^{-1} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}.$$

Since $\Lambda^n$ does not converge, $\lim_{n\to\infty} \mathbf{P}^n$ does not exist. But

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} \Lambda^n = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

so that

$$\lim_{n\to\infty}\frac{1}{n}\sum_{k=0}^{n-1}\mathbf{P}^n = \mathbf{W}\left[\lim_{n\to\infty}\frac{1}{n}\sum_{k=0}^{n-1}\Lambda^n\right]\mathbf{W}^{-1} = \begin{bmatrix}\frac{1}{2} & \frac{1}{2}\\ \\ \frac{1}{2} & \frac{1}{2}\end{bmatrix}.$$

The following important result (which we state without proof) will be fundamental in Chapter **??**. **(Where do we use this?)**

**Theorem 4.4.** If the Markov chain corresponding to the transition matrix $P$ has $k$ closed classes, then the eigenvalue 1 has multiplicity $k$ and $k$ independent eigenvectors.

**(Check what we actually need)** The consequences of this theorem are that:

1. When a Markov chain is unichain or recurrent, the eigenvalue 1 has multiplicity 1 and (right) eigenvector $\mathbf{e} = (1, \ldots, 1)^T$.

2. The space spanned by the right eigenvectors of $\mathbf{P}$ is equivalent to the null space of $\mathbf{I}-\mathbf{P}$. Thus the matrix $\mathbf{I}-\mathbf{P}$ has rank $m-k$ and is not invertible. **(Are next two correct and useful?)**

3. Since $(\mathbf{P}^*)^2 = \mathbf{P}^*$, $\mathbf{P}^*$ is a projection matrix. It can be shown **(Show it?)** that $\mathbf{P}^*$ projects a vector in $\Re^m$ onto the null space of $\mathbf{I}-\mathbf{P}$ so that it distinguishes a particular element of the null space of $\mathbf{I}-\mathbf{P}$.

4. The matrix $\mathbf{I}-\mathbf{P}^*$ is the residual of the projection onto the null space of $\mathbf{I}-\mathbf{P}$. That is a vector $\mathbf{r} = \mathbf{P}^*\mathbf{r}+(\mathbf{I}-\mathbf{P}^*)\mathbf{r}$ where the first expression is in the null space of $\mathbf{I}-\mathbf{P}$ and the second is in its complement and orthogonal to the first term. **(Picture?)**

### Absorbing chains

In Chapter **??** we will study models in which Markov chains corresponding to stationary policies can be transformed into a model with $|S-1|$ transient states and 1 absorbing state. That is $\mathbf{P}$ has the form

$$\mathbf{P} = \begin{bmatrix}\mathbf{Q} & \mathbf{R}\\ \mathbf{0} & 1\end{bmatrix} \tag{4.26}$$

where at least one component of the $|S-1|\times 1$ matrix $\mathbf{R}$ is positive. Thus by the above the theorem, 1 is an eigenvalue of $P$ of multiplicity 1. Moreover we have the following important result[11].

[11] See p.223 in Berman and Plemmons [1979].

**Theorem 4.5.** Suppose $\mathbf{P}$ can be partitioned as in (4.26). Then the spectral radius of $Q$ is strictly less than 1, $\mathbf{Q}^n \to \mathbf{0}$ and

$$(\mathbf{I} - \mathbf{Q})^{-1} = \sum_{n=1}^{\infty} \mathbf{Q}^{n-1}. \tag{4.27}$$

The following example illustrates this result.

**Example 4.6.** Let $S = \{s_1, s_2, s_3\}$ and

$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.3 & 0.2 \\ 0.4 & 0.6 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In this case $\mathbf{Q} = \begin{bmatrix} 0.5 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$ and $\mathbf{R} = \begin{bmatrix} 0.2 \\ 0 \end{bmatrix}$. The eigenvalues of $\mathbf{P}$ are $1, 0.9$ and $0.2$ so that as a consequence of (4.22),

$$\mathbf{P}^* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Moreover as noted in Theorem 4.5 the eigenvalues of $\mathbf{Q}$ are 0.9 and 0.2, and

$$(\mathbf{I} - \mathbf{Q})^{-1} = \begin{bmatrix} 5.00 & 3.75 \\ 5.00 & 6.25 \end{bmatrix}.$$

Note that the $(s, j)$th component of $(\mathbf{I} - \mathbf{Q})^{-1}$ can be interpreted as the (finite) expected number of visits to state $j$. Hence starting in $s_1$, the expected number of visits to $s_1$ equals 5 and the expected number of visits to $s_2$ equal 3.75 so on average starting in state $s_1$ the absorbing state will re reached after 8.75 transitions.

The following example provides another illustration of the consequences of Theorem 4.4.

**Example 4.7.** We illustrate this theorem in the context of the following transition matrix

$$\mathbf{P} = \begin{bmatrix} 0.8 & 0.2 & 0 & 0 \\ 0.4 & 0.6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.2 & 0.2 & 0.4 & 0.2 \end{bmatrix}.$$

Observe that the Markov chain corresponding to this matrix has 2 closed classes $C_1 = \{s_1, s_2\}$ and $C_2 = \{s_3\}$ and a transient state $s_4$. Hence we would expect the eigenvalue 1 to have multiplicity 2.

We see that $\mathbf{P}$ has 4 linearly independent eigenvectors so as a result of Theorem 4.2 we can write $\mathbf{P} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1}$ where[a]

$$\mathbf{\Lambda} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.4 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 1 & 0 & 0.2 & 0 \\ 1 & 0 & -0.4 & 0 \\ 0 & 1 & 0 & 0 \\ 0.5 & 0.5 & 0.2 & 1 \end{bmatrix},$$

$$\text{and } \mathbf{W}^{-1} = \begin{bmatrix} 0.667 & 0.333 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1.667 & 1.667 & 0 & 0 \\ 0 & -0.5 & -0.375 & 1 \end{bmatrix}.$$

Letting $\mathbf{L} = \lim_{n\to\infty} \mathbf{\Lambda}^n$, $\mathbf{P}^* = \mathbf{W}\mathbf{L}\mathbf{W}^{-1}$ can be computed as follows:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

to obtain

$$\mathbf{P}^* = \begin{bmatrix} 0.667 & 0.333 & 0 & 0 \\ 0.667 & 0.333 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.333 & 0.1667 & 0.5 & 0 \end{bmatrix}.$$

Note that starting in state $s_4$, the chain ends in $C_1$ or $C_2$ with probability 0.5 and then follows the limiting distribution for that class.

[a] To find the eigenvectors of $\mathbf{P}$ we find the eigenvectors for each closed class and adjust values in transient states to ensure that $\mathbf{P}\mathbf{v} = \lambda\mathbf{v}$ for all states. See Example 4.4.

### 4.3.11 Countable state chains*

Although not used in this book, we include a brief discussion of some distinctions that arise when analyzing countable state Markov chains. Countable state chains arise naturally in queuing models in which the state represents the number of jobs in the queue. In most places in the book we have addressed this challenge by truncating the state space.

This brief section points out some of the challenges when analyzing countable-state Markov chains as the following simple example shows.

**Example 4.8.** This example shows that that when $S$ is countable the limit

$$\frac{1}{N}\lim_{N\to\infty}\sum_{n=0}^{N-1}\mathbf{P}^n \tag{4.28}$$

exists but that $\mathbf{P}^*$ need not be a transition probability matrix. Let $S = \{1, 2, \dots\}$ and consider Markov chain with transition probabilities $p(s+1|s) = 1$, $p(j|s) = 0$ for $j \neq s+1$. That is at each transition the Markov chain moves to the right with probability one.

Clearly all states are transient, aperiodic and $\lim_{n\to\infty}\mathbf{P}^n$ exists so that the limit in (4.28) also exists. But the limiting matrix $\mathbf{P}^*(j|s) = 0$ is not a transition probability matrix since all the "mass went off to infinity".

Hence if there exists a non-zero reward $\mathbf{r}$ the long run average reward *cannot* be represented by $\mathbf{P}^*\mathbf{r}$.

Such a model is not typical of Markov decision process applications in queuing in which good policies (such as control limit policies in admission control models) in which the resulting Markov chain can be partitioned into a finite set of recurrent states and infinite set of transient states.

Note further that there some distinctions with respect to the concepts of recurrence and transience which are defined in terms of random variables describing the behavior of transitions. Define $\nu_s$ to be the number of times a Markov chain visits state $s$. If $s$ is transient $E_s[\nu_s] < \infty$ and if $E_s[\nu_s] = \infty$ if $s$ is recurrent. However recurrence can be further refined.

Let $\tau_s$ denote the time of the *first visit time to state* $s$[12]. For a recurrent state $s$ $P(\tau_s < \infty|X_0 = s) = 1$ and if $s$ is transient, $P(\tau_s < \infty|X_0 = s) < 1$. That is for a recurrent state, the Markov chain returns to its starting state in a finite time with probability one but it doesn't necessarily return to a transient state. However, the notion of recurrence can be further refined: a state $s$ is *positive recurrent* if $E[\tau_s|X_0 = s] < \infty$ and *null recurrent* if $E[\tau_s|X_0 = s] = \infty$. Null recurrence is a curious phenomenon. It refers to a situation in which the chain returns to its starting state with certainty but the expected time to do so may may be infinite.

The important consequence of this is that in a transient or null recurrent class, the limiting probabilities are 0, while in a positive recurrent class, the limiting probabilities are non-zero. The following example adopted from in Sennott [1999] (p.293-295) illustrates this distinction.

**Example 4.9.** Consider a single server queue with batch arrivals and deterministic service rate of one job per period. Assume further that jobs are admitted only when

[12]If the chain starts in $s$, $\tau_s$ denotes the first return time.

the server is idle, which occurs when the queue is empty.

To model this let $S = \{0, 1, 2, ..\}$, $p(j|0) = q_j$ for $j \geq 1$, $p(s-1|s) = 1$ for $s \geq 1$ and $p(j|s) = 0$ otherwise. Thus each time the queue is empty, the system jumps to state $j$ with probability $q_j$. We leave it as an exercise to show that:

- All states are recurrent.
- If $\sum_{j=1}^{\infty} jq_j < \infty$, all states are positive recurrent.
- If $\sum_{j=1}^{\infty} jq_j = \infty$, all states are null recurrent.

Thus if the batch sizes are too large, the Markov chain is null recurrent.

## 4.4 Linear Programming

A linear program is an optimization problem that comprises an *objective function* and *constraints*, which are restricted to linear functions of the *decision variables*. Let $\mathbf{c} \in \Re^n$, $\mathbf{x} \in \Re^n$, $\mathbf{A} \in \Re^{m \times n}$ and $\mathbf{b} \in \Re^m$. Consider the following linear program.

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad \forall i = 1, \ldots, m. \end{aligned} \tag{4.29}$$

It has a linear objective function and constraints in the form of linear inequalities with respect to the decision variables $x_1, \ldots, x_n$. The *parameters* $c_j, j = 1, \ldots, n, a_{ij}, i = 1, \ldots, m, j = 1, \ldots, n$, and $b_i, i = 1, \ldots, m$ are fixed and typically derived from data.

We can write the above formulation compactly in vector form:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} \end{aligned} \tag{4.30}$$

The $i$-th row of the $\mathbf{A}$ matrix will be written as a row vector $\mathbf{a}_i^\mathsf{T}$.

Note that many equivalent forms of an LP can be written in the above manner. For example, if the objective is to maximize instead of minimize, we can simply replace $\mathbf{c}$ with $-\mathbf{c}$. Similarly, for "less than or equal to" inequalities, we can multiply $\mathbf{A}$ and $\mathbf{b}$ by $-1$. Equality constraints can also be represented in the above form by noting that for a given $i$, $\mathbf{a}_i^\mathsf{T}\mathbf{x} = b_i$ can be enforced with two constraints: $\mathbf{a}_i^\mathsf{T}\mathbf{x} \geq b_i$ and $-\mathbf{a}_i^\mathsf{T}\mathbf{x} \geq -b_i$. Sign constraints on the variables (e.g., $x_1 \geq 0$ or $x_2 \leq 0$) can be enforced with appropriate choices of $\mathbf{A}$ and $\mathbf{b}$.

A *standard form* linear program is written

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{4.31}$$

where the constraints comprise equality constraints involving $\mathbf{A}$ and $\mathbf{b}$, and non-negativity constraints on the decision variables. Any linear program can be transformed to one in standard form as follows. An unconstrained variable $x_j$ can be replaced by $x_j^+ - x_j^-$, with constraints $x_j^+ \geq 0$ and $x_j^- \geq 0$. An inequality $\mathbf{a}_i^\mathsf{T}\mathbf{x} \geq b_i$ can be written as $\mathbf{a}_i^\mathsf{T}\mathbf{x} - s_i = b_i$, where $s_i \geq 0$. If the inequality is a "less than or equal to", we add $s_i$ instead of subtracting it.

A vector $\mathbf{x}$ that satisfies all constraints is a *feasible solution*. The set of vectors that satisfy the constraints is referred to as the *feasible region*. The feasible region of a linear program is a *polyhedron* by definition, since a polyhedron is defined as the intersection of a finite set of linear inequalities. Among the feasible solutions, the one with the lowest value of $\mathbf{c}^\mathsf{T}\mathbf{x}$, if it exists, is an *optimal solution*, denoted $\mathbf{x}^*$. If a linear program is feasible, but for every real number $r$ there exists a feasible solution such that $\mathbf{c}^\mathsf{T}\mathbf{x} < r$, then we say the problem is *unbounded* or has *unbounded objective function value*. Such an LP may be referred to as having an optimal value of $-\infty$. The following theorem establishes when an LP will have an optimal solution.

**Theorem 4.6.** Every feasible LP with bounded objective function has an optimal solution.

If the feasible region of an LP is non-empty and bounded, then the objective function value cannot go to $-\infty$. Hence, the LP will have an optimal solution.

**Corollary 4.1.** An LP with a non-empty bounded feasible region has an optimal solution.

Consider an LP with the set of constraints $\mathbf{a}_i^\mathsf{T}\mathbf{x} \geq b_i, i = 1, \ldots, m$. If a vector $\mathbf{x}$ satisfies a subset of the constraints at equality $\mathbf{a}_i^\mathsf{T} = b_i, i \in I \subseteq \{1, \ldots, m\}$, and if there are $n$ coefficient vectors in this subset $\{\mathbf{a}_i | i \in I\}$ that are linearly independent, then $\mathbf{x}$ is a *basic feasible solution*. Figure **??** from Chapter 4 illustrates a polyhedron. Basic feasible solutions are the "corner points" of the polyhedron. Basic feasible solutions are important because under mild conditions, optimal solutions to an LP, if they exist, can be found there.

**Theorem 4.7.** If a linear program has an optimal solution and its feasible region contains at least one basic feasible solution, then at least one basic feasible solution is an optimal solution.

The well-known Simplex Algorithm progressively searches basic feasible solutions with improving objective function value until it finds an optimal basic feasible solution or determines that the optimal value is $-\infty$.

Note that there are LPs that have optimal solutions, but no corner points. Consider the feasible region $\{(x_1, x_2) \mid x_2 \geq 0\}$, which is a half space of $\Re^2$. If the objective is

to minimize $x_2$, then all points on the line $(x_1, 0), x_1 \in (-\infty, \infty)$ are optimal but none are corner points.

To guarantee that a polyhedron has a corner point, it is necessary and sufficient that it does not contain a line. That is, there must not be a vector $\mathbf{x}$ and nonzero vector $\mathbf{d}$ such that $\mathbf{x} + \alpha\mathbf{d}$ remains in the polyhedron for all scalars (negative and positive) $\boldsymbol{\alpha}$. Note that a linear program in standard form does not contain a line, since the feasible region is a subset of $\{\mathbf{x} \mid \mathbf{x} \geq \mathbf{0}\}$. Hence, if a linear program in standard form has an optimal solution, there must be one at a basic feasible solution.

### 4.4.1 Duality

Given any linear program, there exists a *dual* linear program. We refer to the former or original linear program as the *primal*. The following pair of linear programs are dual to each other.

$$
\begin{array}{rlrl}
\text{minimize} & \mathbf{c}^\mathsf{T}\mathbf{x} & \text{maximize} & \mathbf{b}^\mathsf{T}\mathbf{y} \\
\text{subject to} & \mathbf{a}_i^\mathsf{T}\mathbf{x} \geq b_i, \quad i \in I_1, & \text{subject to} & y_i \geq 0, \quad i \in I_1, \\
& \mathbf{a}_i^\mathsf{T}\mathbf{x} \leq b_i, \quad i \in I_2, & & y_i \leq 0, \quad i \in I_2, \\
& \mathbf{a}_i^\mathsf{T}\mathbf{x} = b_i, \quad i \in I_3, & & y_i \text{ free}, \quad i \in I_3, \\
& x_j \geq 0, \quad j \in J_1, & & \mathbf{A}_j^\mathsf{T}\mathbf{y} \leq c_j, \quad i \in J_1, \\
& x_j \leq 0, \quad j \in J_2, & & \mathbf{A}_j^\mathsf{T}\mathbf{y} \geq c_j, \quad i \in J_2, \\
& x_j \text{ free}, \quad j \in J_3, & & \mathbf{A}_j^\mathsf{T}\mathbf{y} = c_j, \quad i \in J_3.
\end{array}
$$

Let the minimization problem be the primal. Then the dual problem is a maximization problem. We associate a dual variable with every non-sign constraint in the primal. Whether the dual variable is subject to a sign constraint or is free depends on whether the corresponding primal constraint is an inequality or equality, respectively. Similarly, every primal variable is associated with a non-sign constraint in the dual, and whether the primal variable is sign-constrained or is free determines whether the dual constraint is an inequality or equality, respectively. Notice that the constraints involving $\mathbf{A}$ in the dual use the transpose of the $\mathbf{A}$ matrix. That is, the constraints in the primal are written based on the rows, $\mathbf{a}_i^\mathsf{T}$, of $\mathbf{A}$. However, the constraints in the dual are written based on the columns, $\mathbf{A}_j$, of $\mathbf{A}$. Correspondingly, the parameters $\mathbf{c}$ and $\mathbf{b}$ have switched places. The objective coefficients $\mathbf{c}$ in the primal have become the right hand side parameters in the constraints in the dual, and the right hand side parameters $\mathbf{b}$ in the primal have become the objective coefficients in the dual.

First, it is straightforward to show that the value of the dual (maximization) objective is a lower bound on the value of the primal (minimization) objective. This result is known as *Weak Duality*.

**Theorem 4.8.** Let $\mathbf{x}$ be a feasible solution to the primal and $\mathbf{y}$ be a feasible solution to the dual. Then $\mathbf{b}^\mathsf{T}\mathbf{y} \leq \mathbf{c}^\mathsf{T}\mathbf{x}$.

An immediate consequence of this result is that if there is a primal solution and dual solution that have the same objective function value, they are optimal solutions for their respective problems.

**Corollary 4.2.** Let $\mathbf{x}$ be a feasible solution to the primal and $\mathbf{y}$ be a feasible solution to the dual. If $\mathbf{b}^\mathsf{T}\mathbf{y} = \mathbf{c}^\mathsf{T}\mathbf{x}$, then $\mathbf{x}$ and $\mathbf{y}$ are optimal solutions to the primal and dual, respectively.

A fundamental theorem of linear programming is the theorem of *Strong Duality*, which we state next.

**Theorem 4.9.** Let $\mathbf{x}^*$ be an optimal solution for an LP. Then its dual also has an optimal solution, $\mathbf{y}^*$, and their optimal values are equal.

Finally, we present the *Complementary Slackness* conditions, which provide another set of necessary and sufficient conditions for feasible solutions to the primal and dual to be optimal.

**Theorem 4.10.** Let $\mathbf{x}$ be a feasible solution to the primal and $\mathbf{y}$ be a feasible solution to the dual. They are optimal if and only if

$$y_i(\mathbf{a}_i^\mathsf{T}\mathbf{x} - b_i) = 0, \quad \forall i$$
$$(c_j - \mathbf{A}_j^\mathsf{T}\mathbf{y})x_j = 0, \quad \forall j.$$

# Bibliography


O. Alagoz, L. M. Maillart, A. J. Schaefer, and M. S. Roberts. Determining the acceptance of cadaveric livers using an implicit model of the waiting list. *Operations Research*, 55:24–36, 2007.

K. J. Arrow, D. Blackwell, and M. A. Girshick. Bayes and minimax solutions of sequential decision problems. *Econometrica*, 17:213–244, 1949.

K. J. Arrow, T. Harris, and J. Marschak. Optimal inventory policy. *Econometrica*, 19: 250–272, 1951.

K. J. Arrow, S. Karlin, and H. E. Scarf. *Studies in the Mathematical Theory of Inventory and Production*. Stanford University Press, Stanford, CA, 1958.

A. Berman and R. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, 1979.

D. Bertsimas and R. Shioda. Restaurant revenue management. *Operations Research*, 51:472–486, 2003.

D. Blackwell. Discrete dynamic programming. *Ann. Math. Stat.*, 33:719–726, 1962.

V. Carter and R. E. Machol. Technical note — operations research on football. *Operations Research*, 19:541–544, 1971.

A. Cayley. Mathematical questions with their solutions, no. 4528. *Educational Times*, 23:18, 1875.

T. C. Y. Chan and R. Singal. A Markov Decision Process-based handicap system for tennis. *Journal of Quantitative Analysis in Sports*, 12:179–189, 2016.

T. C. Y. Chan, C. Fernandes, and M. L. Puterman. Points gained in football: Using Markov process-based value functions to assess team performance. *Operations Research*, 69:877–894, 2021.

Y. S. Chow, H. Robbins, and D. Siegmund. *Great Expectations: The theory of optimal stopping*. Houghton-Mifflin, New York, 1971.

C. W. Clark. The lazy adaptable lions: a Markovian model of group foraging. *Animal Behavior*, 35:361–368, 1987.

D. P. de Farias and B. van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51:850–865, 2003.

A. Dvoretsky, J. Kiefer, and J. Wolfowitz. The inventory problem: I. case of known distributions of demand. *Econometrica*, 20:187, 1952.

R.G. Gallagher. *Discrete Stochastic Processes* . LibreTexts, 2013.

G. Gallego and G. van Ryzin. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40:999–1020, 1994.

Y. Goçgun and M. L. Puterman. Dynamic scheduling with due dates and time windows: An application to chemotherapy patient appointment booking. *Health Care Management Science*, 17:60–76, 2014.

M. Hall. The state of goalie pulling in the NHL. `https://hockey-graphs.com/2020/05/18/the-state-of-goalie-pulling-in-the-nhl/`, May 2020. [accessed 14-September-2021].

D. P. Heyman. Optimal operating policies for M/G/1 queueing systems. *Operations Research*, 16:362–382, 1968.

R. Howard. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA, 1960.

T. Hubel and N. Jordan O. Dewhirst J. McNutt A. Wilson et al., J. Myatt. Energy cost and return in african wild dogs and cheetahs. *Nature Communications*, 7:11034, 2016.

S. Karlin. Stochastic models and optimal policies for selling an asset. In K. J. Arrow, S. Karlin, and H. Scarf, editors, *Studies in Applied Probability and Management Science*, pages pp. 148–158. Stanford University Press, Palo Alto, CA, 1962.

E. J. Kelly and P. L. Kennedy. A dynamic stochastic model of mate desertion. *Ecology*, 74:351–366, 1993.

J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Van Nostrand-Reinhold, New York, 1960.

M. Kurt, B. T. Denton, A. J. Schaefer, N. D. Shah, and S. A. Smith. The structure of optimal statin initiation policies for patients with type 2 diabetes. *IIE Transactions on Healthcare Systems Engineering*, 1:49–65, 2011.

M. Mangel and C. W. Clark. Towards a unified foraging theory. *Ecology*, 67:1127–1138, 1986.

D. G. Morrison. On the optimal time to pull the goalie: A poisson model applied to a common strategy used in ice hockey. *TIMS Studies in Management Science*, 4: 67–78, 1976.

P. Naor. On the regulation of queue size by levying tolls. *Econometrica*, 37:15–24, 1969.

J. Patrick, M. L. Puterman, and M. Queyranne. Dynamic multi-priority patient scheduling. *Operations Research*, 56:1507–152, 2008.

E. Porteus. *Foundations of Stochastic Inventory Theory.* Stanford Business Books, 2002.

M. L. Puterman. *Markov Decision Processes:Discrete Stochastic Dynamic Programming.* John Wiley & Sons., 1994.

A. Sauré, J. Patrick, and M. L. Puterman. Optimal multi-appointment scheduling. *European Journal of Operational Research*, 223:573–584, 2012.

L. Sennott. *Stochastic Dynamic Programming and the Control of Queueing Systems* . John Wiley and Sons, 1999.

S. M. Shechter, M. D. Bailey, A. J. Schaefer, and M. S. Roberts. The optimal time to initiate HIV therapy under ordered health states. *Operations Research*, 56:20–33, 2008.

E. Sirot and C. Bernstein. Time sharing between host searching and food searching in parasitoids: state-dependent optimal strategies. *Behavioral Ecology*, 7:189–194, 1996.

B. C. Smith, J. F. Leimkuhler, and R. M. Darrow. Yield management at american airlines. *Interfaces*, 22:8–31, 1992.

G.L. Smuts. Diet of lions and spotted hyenas assessed from stomach contents. *S. Afr. J. Wildl. Res.*, 9:19–25, 1979.

M. J. Sobel. Optimal average-cost policy for a queue with start-up and shut-down costs. *Operations Research*, 17:145–162, 1969.

D. Stengos and L. C. Thomas. The blast furnaces problem. *European Journal of Operational Research*, 4:330–336, 1980.

G. Strang. *Introduction to Linear Algebra, 6th edition.* Wellesley-Cambridge Press, 2023.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An introduction, second edition.* MIT Press, Cambridge, MA, 2018.

K. Talluri and G. van Ryzin. *The Theory and Practice of Revenue Management*. Springer US, 2004.

A. Wald. *Sequential Analysis*. John Wiley & Sons, New York, 1947.

A. Wald and J. Wolfowitz. Optimal character of the sequential probability ratio test. *Ann. of Math.. Stat.*, 19:326–339, 1948.

A. Washburn. Still more on pulling the goalie. *Interfaces*, 21:59–64, 1991.

M. Yadin and P. Naor. On queueing systems with variable service capacities. *Naval Research Logistics Quarterly*, 14:43–53, 1967.

# Chapter 5

# Index

1. xx