Chapter 9

Basic Decision Theory

This chapter serves as a building block for modeling and solving planning problems that involve more than one decision maker. The focus is on making a single decision in the presence of other decision makers that may interfere with the outcome. The planning problems in Chapters 10 to 12 will be viewed as a sequence of decision-making problems. The ideas presented in this chapter can be viewed as making a one-stage plan. With respect to Chapter 2, the present chapter reduces the number of stages down to one and then introduces more sophisticated ways to model a single stage. Upon returning to multiple stages in Chapter 10, it will quickly be seen that many algorithms from Chapter 2 extend nicely to incorporate the decision-theoretic concepts of this chapter.

Since there is no information to carry across stages, there will be no need for a state space. Instead of designing a *plan* for a *robot*, in this chapter we will refer to designing a *strategy* for a *decision maker* (DM). The *planning problem* reduces down to a *decision-making problem*. In later chapters, which describe sequential decision making, planning terminology will once again be used. It does not seem appropriate yet in this chapter because making a single decision appears too degenerate to be referred to as planning.

A consistent theme throughout Part III will be the interaction of multiple DMs. In addition to the primary DM, which has been referred to as the robot, there will be one or more other DMs that cannot be predicted or controlled by the robot. A special DM called *nature* will be used as a universal way to model uncertainties. Nature will usually be fictitious in the sense that it is not a true entity that makes intelligent, rational decisions for its own benefit. The introduction of nature merely serves as a convenient modeling tool to express many different forms of uncertainty. In some settings, however, the DMs may actually be intelligent opponents who make decisions out of their own self-interest. This leads to *game theory*, in which all decision makers (including the robot) can be called *players*.

Section 9.1 provides some basic review and perspective that will help in understanding and relating later concepts in the chapter. Section 9.2 covers making a single decision under uncertainty, which is typically referred to as *decision theory*. Sections 9.3 and 9.4 address *game theory*, in which two or more DMs make their

decisions simultaneously and have conflicting interests. In *zero-sum game theory*, which is covered in Section 9.3, there are two DMs that have diametrically opposed interests. In *nonzero-sum game theory*, covered in Section 9.4, any number of DMs come together to form a *noncooperative game*, in which any degree of conflict or competition is allowable among them. Section 9.5 concludes the chapter by covering justifications and criticisms of the general models formulated in this chapter. It useful when trying to apply decision-theoretic models to planning problems in general.

This chapter was written without any strong dependencies on Part II. In fact, even the concepts from Chapter 2 are not needed because there are no stages or state spaces. Occasional references to Part II will be given, but these are not vital to the understanding. Most of the focus in this chapter is on discrete spaces.

9.1 Preliminary Concepts

9.1.1 Optimization

9.1.1.1 Optimizing a single objective

Before progressing to complicated decision-making models, first consider the simple case of a single decision maker that must make the best decision. This leads to a familiar *optimization* problem, which is formulated as follows.

Formulation 9.1 (Optimization)

1. A nonempty set U called the *action space*. Each $u \in U$ is referred to as an *action*.

2. A function $L : U \rightarrow \mathbb{R} \cup \{\infty\}$ called the *cost function*.

Compare Formulation 9.1 to Formulation 2.2. State space, X, and state transition concepts are no longer needed because there is only one decision. Since there is no state space, there is also no notion of initial and goal states. A *strategy* simply consists of selecting the best action.

What does it mean to be the "best" action? If U is finite, then the best action, $u^* \in U$ is

$$u^* = \operatorname*{argmin}_{u \in U} \Big\{ L(u) \Big\}. \tag{9.1}$$

If U is infinite, then there are different cases. Suppose that $U = (-1, 1)$ and $L(u) = u$. Which action produces the lowest cost? We would like to declare that -1 is the lowest cost, but $-1 \notin U$. If we had instead defined $U = [-1, 1]$, then this would work. However, if $U = (-1, 1)$ and $L(u) = u$, then there is no action that produces minimum cost. For any action $u \in U$, a second one, $u' \in U$, can always be chosen for which $L(u') < L(u)$. However, if $U = (-1, 1)$ and $L(u) = |u|$, then (9.1) correctly reports that $u = 0$ is the best action. There is no problem in this

case because the minimum occurs in the interior, as opposed to on the boundary of U. In general it is important to be aware that an optimal value may not exist.

There are two ways to fix this frustrating behavior. One is to require that U is a closed set and is bounded (both were defined in Section 4.1). Since closed sets include their boundary, this problem will be avoided. The bounded condition prevents a problem such as optimizing $U = \mathbb{R}$, and $L(u) = u$. What is the best $u \in U$? Smaller and smaller values can be chosen for u to produce a lower cost, even though $\mathbb{R}$ is a closed set.

The alternative way to fix this problem is to define and use the notion of an *infimum*, denoted by inf. This is defined as the largest lower bound that can be placed on the cost. In the case of $U = (-1, 1)$ and $L(u) = u$, this is

$$\inf_{u \in (-1,1)} \Big\{ L(u) \Big\} = -1. \tag{9.2}$$

The only difficulty is that there is no action $u \in U$ that produces this cost. The infimum essentially uses the closure of U to evaluate (9.2). If U happened to be closed already, then u would be included in U. Unbounded problems can also be handled. The infimum for the case of $U = \mathbb{R}$ and $L(u) = u$ is $-\infty$.

As a general rule, if you are not sure which to use, it is safer to write inf in the place were you would use min. The infimum happens to yield the minimum whenever a minimum exists. In addition, it gives a reasonable answer when no minimum exists. It may look embarrassing, however, to use inf in cases where it is obviously not needed (i.e., in the case of a finite U).

It is always possible to make an "upside-down" version of an optimization problem by multiplying L by -1. There is no fundamental change in the result, but sometimes it is more natural to formulate a problem as one of maximization instead of minimization. This will be done, for example, in the discussion of utility theory in Section 9.5.1. In such cases, a *reward function*, R, is defined instead of a cost function. The task is to select an action $u \in U$ that *maximizes* the reward. It will be understood that a maximization problem can easily be converted into a minimization problem by setting $L(u) = -R(u)$ for all $u \in U$. For maximization problems, the infimum can be replaced by the *supremum*, sup, which is the least upper bound on $R(u)$ over all $u \in U$.

For most problems in this book, the selection of an optimal $u \in U$ in a single decision stage is straightforward; planning problems are instead complicated by many other aspects. It is important to realize, however, that optimization itself is an extremely challenging if U and L are complicated. For example, U may be finite but extremely large, or U may be a high-dimensional (e.g., 1000) subset of $\mathbb{R}^n$. Also, the cost function may be extremely difficult or even impossible to express in a simple closed form. If the function is simple enough, then standard calculus tools based on first and second derivatives may apply. It most real-world applications, however, more sophisticated techniques are needed. Many involve a form of gradient descent and therefore only ensure that a local minimum is found. In many cases, sampling-based techniques are needed. In fact, many of the

sampling ideas of Section 5.2, such as dispersion, were developed in the context of optimization. For some classes of problems, combinatorial solutions may exist. For example, *linear programming* involves finding the min or max of a collection of linear functions, and many combinatorial approaches exist [259, 264, 664, 731]. This optimization problem will appear in Section 9.4.

Given the importance of sampling-based and combinatorial methods in optimization, there are interesting parallels to motion planning. Chapters 5 and 6 each followed these two philosophies, respectively. Optimal motion planning actually corresponds to an optimization problem on the space of paths, which is extremely difficult to characterize. In some special cases, as in Section 6.2.4, it is possible to find optimal solutions, but in general, such problems are extremely challenging. *Calculus of variations* is a general approach for addressing optimization problems over a space of paths that must satisfy differential constraints [841]; this will be covered in Section 13.4.1.

9.1.1.2 Multiobjective optimization

Suppose that there is a collection of cost functions, each of which evaluates an action. This leads to a generalization of Formulation 9.1 to multiobjective optimization.

Formulation 9.2 (Multiobjective Optimization)

1. A nonempty set U called the *action space*. Each $u \in U$ is referred to as an *action*.

2. A vector-valued *cost function* of the form $L : U \rightarrow \mathbb{R}^d$ for some integer d. If desired, ∞ may also be allowed for any of the cost components.

A version of this problem was considered in Section 7.7.2, which involved the optimal coordination of multiple robots. Two actions, u and u', are called *equivalent* if $L(u) = L(u')$. An action u is said to *dominate* an action u' if they are not equivalent and $L_i(u) \leq L_i(u')$ for all i such that $1 \leq i \leq d$. This defines a partial ordering, $\leq$, on the set of actions. Note that many actions may be *incomparable*. An action is called *Pareto optimal* if it is not dominated by any others. This means that it is minimal with respect to the partial ordering.

Example 9.1 (Simple Example of Pareto Optimality) Suppose that $U = \{1, 2, 3, 4, 5\}$ and $d = 2$. The costs are assigned as $L(1) = (4, 0)$, $L(2) = (3, 3)$, $L(3) = (2, 2)$, $L(4) = (5, 7)$, and $L(5) = (9, 0)$. The actions 2, 4, and 5 can be eliminated because they are dominated by other actions. For example, $(3, 3)$ is dominated by $(2, 2)$; hence, action $u = 3$ is preferable to $u = 2$. The remaining two actions, $u = 1$ and $u = 3$, are Pareto optimal. ■

Based on this simple example, the notion of Pareto optimality seems mostly aimed at discarding dominated actions. Although there may be multiple Pareto-optimal solutions, it at least narrows down U to a collection of the best alternatives.

Example 9.2 (Pennsylvania Turnpike) Imagine driving across the state of Pennsylvania and being confronted with the Pennsylvania Turnpike, which is a toll highway that once posted threatening signs about speed limits and the according fines for speeding. Let $U = \{50, 51, \ldots, 100\}$ represent possible integer speeds, expressed in miles per hour (mph). A posted sign indicates that the speeding fines are 1) \$50 for being caught driving between 56 and 65 mph, 2) \$100 for being caught between 66 and 75, 3) \$200 between 76 and 85, and 4) \$500 between 86 and 100. Beyond 100 mph, it is assumed that the penalty includes jail time, which is so severe that it will not be considered.

The two criteria for a driver are 1) the time to cross the state, and 2) the amount of money spent on tickets. It is assumed that you will be caught violating the speed limit. The goal is to minimize both. What are the resulting Pareto-optimal driving speeds? Compare driving 56 mph to driving 57 mph. Both cost the same amount of money, but driving 57 mph takes less time. Therefore, 57 mph dominates 56 mph. In fact, 65 mph dominates all speeds down to 56 mph because the cost is the same, and it reduces the time the most. Based on this argument, the Pareto-optimal driving speeds are 55, 65, 75, 85, and 100. It is up to the individual drivers to decide on the particular best action for them; however, it is clear that no speeds outside of the Pareto-optimal set are sensible. ■

The following example illustrates the main frustration with Pareto optimality. Removing nondominated solutions may not be useful enough. In come cases, there may even be a continuum of Pareto-optimal solutions. Therefore, the Pareto-optimal concept is not always useful. Its value depends on the particular application.

Example 9.3 (A Continuum of Pareto-Optimal Solutions) Let $U = [0, 1]$ and $d = 2$. Let $L(u) = (u, 1 - u)$. In this case, every element of U is Pareto optimal. This can be seen by noting that a slight reduction in one criterion causes an increase in the other. Thus, any two actions are incomparable. ■

9.1.2 Probability Theory Review

This section reviews some basic probability concepts and introduces notation that will be used throughout Part III.

Probability space A *probability space* is a three-tuple, $(S, \mathcal{F}, P)$, in which the three components are

1. **Sample space:** A nonempty set S called the *sample space*, which represents all possible outcomes.

2. **Event space:** A collection $\mathcal{F}$ of subsets of S, called the *event space*. If S is discrete, then usually $\mathcal{F} = \text{pow}(S)$. If S is continuous, then $\mathcal{F}$ is usually a sigma-algebra on S, as defined in Section 5.1.3.

3. **Probability function:** A function, $P : \mathcal{F} \rightarrow \mathbb{R}$, that assigns probabilities to the events in $\mathcal{F}$. This will sometimes be referred to as a *probability distribution* over S.

The probability function, P, must satisfy several basic axioms:

1. $P(E) \geq 0$ for all $E \in \mathcal{F}$.

2. $P(S) = 1$.

3. $P(E \cup F) = P(E) + P(F)$ if $E \cap F = \emptyset$, for all $E, F \in \mathcal{F}$.

If S is discrete, then the definition of P over all of $\mathcal{F}$ can be inferred from its definition on single elements of S by using the axioms. It is common in this case to write $P(s)$ for some $s \in S$, which is slightly abusive because s is not an event. It technically should be $P(\{s\})$ for some $\{s\} \in \mathcal{F}$.

Example 9.4 (Tossing a Die) Consider tossing a six-sided cube or die that has numbers 1 to 6 painted on its sides. When the die comes to rest, it will always show one number. In this case, $S = \{1, 2, 3, 4, 5, 6\}$ is the sample space. The event space is $\text{pow}(S)$, which is all 2^6 subsets of S. Suppose that the probability function is assigned to indicate that all numbers are equally likely. For any individual $s \in S$, $P(\{s\}) = 1/6$. The events include all subsets so that any probability statement can be formulated. For example, what is the probability that an even number is obtained? The event $E = \{2, 4, 6\}$ has probability $P(E) = 1/2$ of occurring. ■

The third probability axiom looks similar to the last axiom in the definition of a measure space in Section 5.1.3. In fact, P is technically a special kind of measure space as mentioned in Example 5.12. If S is continuous, however, this measure cannot be captured by defining probabilities over the singleton sets. The probabilities of singleton sets are usually zero. Instead, a *probability density function*, $p : S \rightarrow \mathbb{R}$, is used to define the probability measure. The probability function, P, for any event $E \in \mathcal{F}$ can then be determined via integration:

$$P(E) = \int_E p(x) dx, \tag{9.3}$$

in which $x \in E$ is the variable of integration. Intuitively, P indicates the total probability mass that accumulates over E.

Conditional probability A *conditional probability* is expressed as $P(E|F)$ for any two events $E, F \in \mathcal{F}$ and is called the "probability of E, given F." Its definition is

$$P(E|F) = \frac{P(E \cap F)}{P(F)}. \tag{9.4}$$

Two events, E and F, are called *independent* if and only if $P(E \cap F) = P(E)P(F)$; otherwise, they are called *dependent*. An important and sometimes misleading concept is *conditional independence*. Consider some third event, $G \in \mathcal{F}$. It might be the case that E and F are dependent, but when G is given, they become independent. Thus, $P(E \cap F) \neq P(E)P(F)$; however, $P(E \cap F|G) = P(E|G)P(F|G)$. Such examples occur frequently in practice. For example, E might indicate someone's height, and F is their reading level. These will generally be dependent events because children are generally shorter and have a lower reading level. If we are given the person's age as an event G, then height is no longer important. It seems intuitive that there should be no correlation between height and reading level once the age is given.

The definition of conditional probability, (9.4), imposes the constraint that

$$P(E \cap F) = P(F)P(E|F) = P(E)P(F|E), \tag{9.5}$$

which nicely relates $P(E|F)$ to $P(F|E)$. This results in *Bayes' rule*, which is a convenient way to swap E and F:

$$P(F|E) = \frac{P(E|F)P(F)}{P(E)}. \tag{9.6}$$

The probability distribution, $P(F)$, is referred to as the *prior*, and $P(F|E)$ is the *posterior*. These terms indicate that the probabilities come before and after E is considered, respectively.

If all probabilities are conditioned on some event, $G \in \mathcal{F}$, then *conditional Bayes' rule* arises, which only differs from (9.6) by placing the condition G on all probabilities:

$$P(F|E, G) = \frac{P(E|F, G)P(F|G)}{P(E|G)}. \tag{9.7}$$

Marginalization Let the events $F_1, F_2, \ldots, F_n$ be any partition of S. The probability of an event E can be obtained through *marginalization* as

$$P(E) = \sum_{i=1}^{n} P(E|F_i)P(F_i). \tag{9.8}$$

One of the most useful applications of marginalization is in the denominator of Bayes' rule. A substitution of (9.8) into the denominator of (9.6) yields

$$P(F|E) = \frac{P(E|F)P(F)}{\sum_{i=1}^{n} P(E|F_i)P(F_i)}. \tag{9.9}$$

This form is sometimes easier to work with because $P(E)$ appears to be eliminated.

Random variables Assume that a probability space $(S, \mathcal{F}, P)$ is given. A *random variable*[1] X is a function that maps S into $\mathbb{R}$. Thus, X assigns a real value to every element of the sample space. This enables statistics to be conveniently computed over a probability space. If S is already a subset of $\mathbb{R}$, X may by default represent the identity function.

Expectation The *expectation* or *expected value* of a random variable X is denoted by $E[X]$. It can be considered as a kind of weighted average for X, in which the weights are obtained from the probability distribution. If S is discrete, then

$$E[X] = \sum_{s \in S} X(s)P(s). \tag{9.10}$$

If S is continuous, then[2]

$$E[X] = \int_S X(s)p(s)ds. \tag{9.11}$$

One can then define *conditional expectation*, which applies a given condition to the probability distribution. For example, if S is discrete and an event F is given, then

$$E[X|F] = \sum_{s \in S} X(s)P(s|F). \tag{9.12}$$

Example 9.5 (Tossing Dice) Returning to Example 9.4, the elements of S are already real numbers. Hence, a random variable X can be defined by simply letting $X(s) = s$. Using (9.11), the expected value, $E[X]$, is 3.5. Note that the expected value is not necessarily a value that is "expected" in practice. It is impossible to actually obtain 3.5, even though it is not contained in S. Suppose that the expected value of X is desired only over trials that result in numbers greater then 3. This can be described by the event $F = \{4, 5, 6\}$. Using conditional expectation, (9.12), the expected value is $E[X|F] = 5$.

Now consider tossing two dice in succession. Each element $s \in S$ is expressed as $s = (i, j)$ in which $i, j \in \{1, 2, 3, 4, 5, 6\}$. Since $S \not\subset \mathbb{R}$, the random variable needs to be slightly more interesting. One common approach is to count the sum of the dice, which yields $X(s) = i + j$ for any $s \in S$. In this case, $E[X] = 7$. ■

[1] This is a terrible name, which often causes confusion. A random variable is not "random," nor is it a "variable." It is simply a function, $X : S \to \mathbb{R}$. To make matters worse, a capital letter is usually used to denote it, whereas lowercase letters are usually used to denote functions.

[2] Using the language of measure theory, both definitions are just special cases of the Lebesgue integral. Measure theory nicely unifies discrete and continuous probability theory, thereby avoiding the specification of separate cases. See [346, 546, 836].

9.1.3 Randomized Strategies

Up until now, any actions taken in a plan have been *deterministic*. The plans in Chapter 2 specified actions with complete certainty. Formulation 9.1 was solved by specifying the best action. It can be viewed as a *strategy* that trivially makes the same decision every time.

In some applications, the decision maker may not want to be predictable. To achieve this, randomization can be incorporated into the strategy. If U is discrete, a *randomized strategy*, w, is specified by a probability distribution, $P(u)$, over U. Let W denote the set of all possible randomized strategies. When the strategy is applied, an action $u \in U$ is chosen by sampling according to the probability distribution, $P(u)$. We now have to make a clear distinction between *defining the strategy* and *applying the strategy*. So far, the two have been equivalent; however, a randomized strategy must be *executed* to determine the resulting action. If the strategy is executed repeatedly, it is assumed that each trial is independent of the actions obtained in previous trials. In other words, $P(u_k|u_i) = P(u_k)$, in which $P(u_k|u_i)$ represents the probability that the strategy chooses action u_k in trial k, given that u_i was chosen in trial i for some $i < k$. If U is continuous, then a randomized strategy may be specified by a probability density function, $p(u)$. In decision-theory and game-theory literature, deterministic and randomized strategies are often referred to as *pure* and *mixed*, respectively.

Example 9.6 (Basing Decisions on a Coin Toss) Let $U = \{a, b\}$. A randomized strategy w can be defined as

1. Flip a fair coin, which has two possible outcomes: heads (H) or tails (T).
2. If the outcome is H, choose a; otherwise, choose b.

Since the coin is fair, w is defined by assigning $P(a) = P(b) = 1/2$. Each time the strategy is applied, it not known what action will be chosen. Over many trials, however, it converges to choosing a half of the time. ■

A deterministic strategy can always be viewed as a special case of a randomized strategy, if you are not bothered by events that have probability zero. A deterministic strategy, $u_i \in U$, can be simulated by a random strategy by assigning $P(u) = 1$ if $u = u_i$, and $P(u) = 0$ otherwise. Only with probability zero can different actions be chosen (possible, but not probable!).

Imagine using a randomized strategy to solve a problem expressed using Formulation 9.1. The first difficulty appears to be that the cost cannot be predicted. If the strategy is applied numerous times, then we can define the average cost. As the number of times tends to infinity, this average would converge to the expected cost, denoted by $\bar{L}(w)$, if L is treated as a random variable (in addition to the cost function). If U is discrete, the expected cost of a randomized strategy w is

$$\bar{L}(w) = \sum_{u \in U} L(u)P(u) = \sum_{u \in U} L(u)w_i, \tag{9.13}$$

in which w_i is the component of w corresponding to the particular $u \in U$.

An interesting question is whether there exists some $w \in W$ such that $\bar{L}(w) < L(u)$, for all $u \in U$. In other words, do there exist randomized strategies that are better than all deterministic strategies, using Formulation 9.1? The answer is *no* because the best strategy is always to assign probability one to the action, u^*, that minimizes L. This is equivalent to using a deterministic strategy. If there are two or more actions that obtain the optimal cost, then a randomized strategy could arbitrarily distribute all of the probability mass between these. However, there would be no further reduction in cost. Therefore, randomization seems pointless in this context, unless there are other considerations.

One important example in which a randomized strategy is of critical importance is when making decisions in competition with an intelligent adversary. If the problem is repeated many times, an opponent could easily learn any deterministic strategy. Randomization can be used to weaken the prediction capabilities of an opponent. This idea will be used in Section 9.3 to obtain better ways to play zero-sum games.

Following is an example that illustrates the advantage of randomization when repeatedly playing against an intelligent opponent.

Example 9.7 (Matching Pennies) Consider a game in which two players repeatedly play a simple game of placing pennies on the table. In each trial, the players must place their coins simultaneously with either heads (H) facing up or tails (T) facing up. Let a two-letter string denote the outcome. If the outcome is HH or TT (the players choose the same), then Player 1 pays Player 2 one Peso; if the outcome is HT or TH, then Player 2 pays Player 1 one Peso. What happens if Player 1 uses a deterministic strategy? If Player 2 can determine the strategy, then he can choose his strategy so that he always wins the game. However, if Player 1 chooses the best randomized strategy, then he can expect at best to break even on average. What randomized strategy achieves this?

A generalization of this to three actions is the famous game of Rock-Paper-Scissors [958]. If you want to design a computer program that repeatedly plays this game against smart opponents, it seems best to incorporate randomization. ■

9.2 A Game Against Nature

9.2.1 Modeling Nature

For the first time in this book, uncertainty will be directly modeled. There are two DMs:

> **Robot:** This is the name given to the primary DM throughout the book. So far, there has been only one DM. Now that there are two, the name is

more important because it will be used to distinguish the DMs from each other.

Nature: This DM is a mysterious force that is unpredictable to the robot. It has its own set of actions, and it can choose them in a way that interferes with the achievements of the robot. Nature can be considered as a synthetic DM that is constructed for the purposes of modeling uncertainty in the decision-making or planning process.

Imagine that the robot and nature each make a decision. Each has a set of actions to choose from. Suppose that the cost depends on which actions are chosen by each. The cost still represents the effect of the outcome on the robot; however, the robot must now take into account the influence of nature on the cost. Since nature is unpredictable, the robot must formulate a model of its behavior. Assume that the robot has a set, U, of actions, as before. It is now assumed that nature also has a set of actions. This is referred to as the *nature action space* and is denoted by Θ. A *nature action* is denoted as $\theta \in \Theta$. It now seems appropriate to call U the *robot action space*; however, for convenience, it will often be referred to as the *action space*, in which the *robot* is implied.

This leads to the following formulation, which extends Formulation 9.1.

Formulation 9.3 (A Game Against Nature)

1. A nonempty set U called the *(robot) action space*. Each $u \in U$ is referred to as an *action*.

2. A nonempty set Θ called the *nature action space*. Each $\theta \in \Theta$ is referred to as a *nature action*.

3. A function $L : U \times \Theta \rightarrow \mathbb{R} \cup \{\infty\}$, called the *cost function*.

The cost function, L, now depends on $u \in U$ and $\theta \in \Theta$. If U and Θ are finite, then it is convenient to specify L as a $|U| \times |\Theta|$ matrix called the *cost matrix*.

Example 9.8 (A Simple Game Against Nature) Suppose that U and Θ each contain three actions. This results in nine possible outcomes, which can be specified by the following cost matrix:

		Θ		
		1	-1	0
U		-1	2	-2
		2	-1	1

The robot action, $u \in U$, selects a row, and the nature action, $\theta \in \Theta$, selects a column. The resulting cost, $L(u, \theta)$, is given by the corresponding matrix entry. ■

In Formulation 9.3, it appears that both DMs act at the same time; nature does not know the robot action before deciding. In many contexts, nature may know the robot action. In this case, a different nature action space can be defined for every $u \in U$. This generalizes Formulation 9.3 to obtain:

Formulation 9.4 (Nature Knows the Robot Action)

1. A nonempty set U called the *action space*. Each $u \in U$ is referred to as an *action*.

2. For each $u \in U$, a nonempty set $\Theta(u)$ called the *nature action space*.

3. A function $L : U \times \Theta \rightarrow \mathbb{R} \cup \{\infty\}$, called the *cost function*.

If the robot chooses an action $u \in U$, then nature chooses from $\Theta(u)$.

9.2.2 Nondeterministic vs. Probabilistic Models

What is the best decision for the robot, given that it is engaged in a game against nature? This depends on what information the robot has regarding how nature chooses its actions. It will always be assumed that the robot does not know the precise nature action to be chosen; otherwise, it is pointless to define nature. Two alternative models that the robot can use for nature will be considered. From the robot's perspective, the possible models are

Nondeterministic: I have no idea what nature will do.

Probabilistic: I have been observing nature and gathering statistics.

Under both models, it is assumed that the robot knows Θ in Formulation 9.3 or $\Theta(u)$ for all $u \in U$ in Formulation 9.4. The nondeterministic and probabilistic terminology are borrowed from Erdmann [313]. In some literature, the term *possibilistic* is used instead of *nondeterministic*. This is an excellent term, but it is unfortunately too similar to *probabilistic* in English.

Assume first that Formulation 9.3 is used and that U and Θ are finite. Under the nondeterministic model, there is no additional information. One reasonable approach in this case is to make a decision by assuming the worst. It can even be imagined that nature knows what action the robot will take, and it will spitefully choose a nature action that drives the cost as high as possible. This pessimistic view is sometimes humorously referred to as Murphy's Law ("If anything can go wrong, it will.") [111] or Sod's Law. In this case, the best action, $u^* \in U$, is selected as

$$u^* = \operatorname*{argmin}_{u \in U} \Big\{ \max_{\theta \in \Theta} \Big\{ L(u, \theta) \Big\} \Big\}. \tag{9.14}$$

The action u^* is the lowest cost choice using *worst-case analysis*. This is sometimes referred to as a *minimax* solution because of the min and max in (9.14). If U or

Θ is infinite, then the min or max may not exist and should be replaced by inf or sup, respectively.

Worst-case analysis may seem too pessimistic in some applications. Perhaps the assumption that all actions in Θ are equally likely may be preferable. This can be handled as a special case of the probabilistic model, which is described next.

Under the probabilistic model, it is assumed that the robot has gathered enough data to reliably estimate $P(\theta)$ (or $p(\theta)$ if Θ is continuous). In this case, it is imagined that nature applies a randomized strategy, as defined in Section 9.1.3. It assumed that the applied nature actions have been observed over many trials, and in the future they will continue to be chosen in the same manner, as predicted by the distribution $P(\theta)$. Instead of worst-case analysis, *expected-case analysis* is used. This optimizes the average cost to be received over numerous independent trials. In this case, the best action, $u^* \in U$, is

$$u^* = \operatorname*{argmin}_{u \in U} \Big\{ E_\theta \Big[L(u, \theta) \Big] \Big\}, \tag{9.15}$$

in which E_θ indicates that the expectation is taken according to the probability distribution (or density) over θ. Since Θ and $P(\theta)$ together form a probability space, $L(u, \theta)$ can be considered as a random variable for each value of u (it assigns a real value to each element of the sample space).[3] Using $P(\theta)$, the expectation in (9.15) can be expressed as

$$E_\theta[L(u, \theta)] = \sum_{\theta \in \Theta} L(u, \theta) P(\theta). \tag{9.16}$$

Example 9.9 (Nondeterministic vs. Probabilistic) Return to Example 9.8. Let $U = \{u_1, u_2, u_3\}$ represent the robot actions, and let $\Theta = \{\theta_1, \theta_2, \theta_3\}$ represent the nature actions.

Under the nondeterministic model of nature, $u^* = u_1$, which results in $L(u^*, \theta) = 1$ in the worst case using (9.14). Under the probabilistic model, let $P(\theta_1) = 1/5$, $P(\theta_2) = 1/5$, and $P(\theta_3) = 3/5$. To find the optimal action, (9.15) can be used. This involves computing the expected cost for each action:

$$\begin{aligned} E_\theta[L(u_1, \theta)] &= (1)1/5 + (-1)1/5 + (0)3/5 = 0 \\ E_\theta[L(u_2, \theta)] &= (-1)1/5 + (2)1/5 + (-2)3/5 = -1 \\ E_\theta[L(u_3, \theta)] &= (2)1/5 + (-1)1/5 + (1)3/5 = 4/5. \end{aligned} \tag{9.17}$$

The best action is $u^* = u_2$, which produces the lowest expected cost, -1.

If the probability distribution had instead been $P = [1/10 \;\; 4/5 \;\; 1/10]$, then $u^* = u_1$ would have been obtained. Hence the best decision depends on $P(\theta)$; if this information is statistically valid, then it enables more informed decisions to be made. If such information is not available, then the nondeterministic model may be more suitable.

[3] Alternatively, a random variable may be defined over $U \times \Theta$, and conditional expectation would be taken, in which u is given.

It is possible, however, to assign $P(\theta)$ as a uniform distribution in the absence of data. This means that all nature actions are equally likely; however, conclusions based on this are dangerous; see Section 9.5. ■

In Formulation 9.4, the nature action space $\Theta(u)$ depends on $u \in U$, the robot action. Under the nondeterministic model, (9.14) simply becomes

$$u^* = \operatorname*{argmin}_{u \in U} \Big\{ \max_{\theta \in \Theta(u)} L(u, \theta) \Big\}. \tag{9.18}$$

Unfortunately, these problems do not have a nice matrix representation because the size of $\Theta(u)$ can vary for different $u \in U$. In the probabilistic case, $P(\theta)$ is replaced by a conditional probability distribution $P(\theta|u)$. Estimating this distribution requires observing numerous independent trials for each possible $u \in U$. The behavior of nature can now depend on the robot action; however, nature is still characterized by a randomized strategy. It does not adapt its strategy across multiple trials. The expectation in (9.16) now becomes

$$E_\theta \Big[L(u, \theta) \Big] = \sum_{\theta \in \Theta(u)} L(u, \theta) P(\theta|u), \tag{9.19}$$

which replaces $P(\theta)$ by $P(\theta|u)$.

Regret It is important to note that the models presented here are not the only accepted ways to make good decisions. In game theory, the key idea is to minimize "regret." This is the feeling you get after making a bad decision and wishing that you could change it after the game is finished. Suppose that after you choose some $u \in U$, you are told which $\theta \in \Theta$ was applied by nature. The regret is the amount of cost that you could have saved by picking a different action, given the nature action that was applied.

For each combination of $u \in U$ and $\theta \in \Theta$, the *regret*, T, is defined as

$$T(u, \theta) = \max_{u' \in U} \Big\{ L(u, \theta) - L(u', \theta) \Big\}. \tag{9.20}$$

For Formulation 9.3, if U and Θ are finite, then a $|\Theta| \times |U|$ *regret matrix* can be defined.

Suppose that minimizing regret is the primary concern, as opposed to the actual cost received. Under the nondeterministic model, the action that minimizes the worst-case regret is

$$u^* = \operatorname*{argmin}_{u \in U} \Big\{ \max_{\theta \in \Theta} \Big\{ T(u, \theta) \Big\} \Big\}. \tag{9.21}$$

In the probabilistic model, the action that minimizes the expected regret is

$$u^* = \operatorname*{argmin}_{u \in U} \Big\{ E_\theta \Big[T(u, \theta) \Big] \Big\}. \tag{9.22}$$

The only difference with respect to (9.14) and (9.15) is that L has been replaced by T. In Section 9.3.2, regret will be discussed in more detail because it forms the basis of optimality concepts in game theory.

Example 9.10 (Regret Matrix) The regret matrix for Example 9.8 is

		Θ	
	2	0	2
U	0	3	0
	3	0	3

Using the nondeterministic model, $u^* = u_1$, which results in a worst-case regret of 2 using (9.21). Under the probabilistic model, let $P(\theta_1) = P(\theta_2) = P(\theta_3) = 1/3$. In this case, $u^* = u_1$, which yields the optimal expected regret, calculated as 1 using (9.22).

9.2.3 Making Use of Observations

Formulations 9.3 and 9.4 do not allow the robot to receive any information (other than L) prior to making its decision. Now suppose that the robot has a sensor that it can check just prior to choosing the best action. This sensor provides an *observation* or measurement that contains information about which nature action might be chosen. In some contexts, the nature action can be imagined as a kind of *state* that has already been selected. The observation then provides information about this. For example, nature might select the current temperature in Bangkok. An observation could correspond to a thermometer in Bangkok that takes a reading.

Formulating the problem Let Y denote the *observation space*, which is the set of all possible observations, $y \in Y$. For convenience, suppose that Y, U, and Θ are all discrete. It will be assumed as part of the model that some constraints on θ are known once y is given. Under the nondeterministic model a set $Y(\theta) \subseteq Y$ is specified for every $\theta \in \Theta$. The set $Y(\theta)$ indicates the possible observations, given that the nature action is θ. Under the probabilistic model a conditional probability distribution, $P(y|\theta)$, is specified. Examples of sensing models will be given in Section 9.2.4. Many others appear in Sections 11.1.1 and 11.5.1, although they are expressed with respect to a state space X that reduces to Θ in this section. As before, the probabilistic case also requires a prior distribution, $P(\Theta)$, to be given. This results in the following formulation.

Formulation 9.5 (A Game Against Nature with an Observation)

1. A finite, nonempty set U called the *action space*. Each $u \in U$ is referred to as an *action*.

2. A finite, nonempty set Θ called the *nature action space*.

3. A finite, nonempty set Y called the *observation space*.

4. A set $Y(\theta) \subseteq Y$ or probability distribution $P(y|\theta)$ specified for every $\theta \in \Theta$. This indicates which observations are possible or probable, respectively, if θ is the nature action. In the probabilistic case a prior, $P(\theta)$, must also be specified.

5. A function $L : U \times \Theta \rightarrow \mathbb{R} \cup \{\infty\}$, called the *cost function*.

Consider solving Formulation 9.5. A strategy is now more complicated than simply specifying an action because we want to completely characterize the behavior of the robot before the observation has been received. This is accomplished by defining a *strategy* as a function, $\pi : Y \rightarrow U$. For each possible observation, $y \in Y$, the strategy provides an action. We now want to search the space of possible strategies to find the one that makes the best decisions over all possible observations. In this section, Y is actually a special case of an information space, which is the main topic of Chapters 11 and 12. Eventually, a strategy (or plan) will be conditioned on an information state, which generalizes an observation.

Optimal strategies Now consider finding the optimal strategy, denoted by π^*, under the nondeterministic model. The sets $Y(\theta)$ for each $\theta \in \Theta$ must be used to determine which nature actions are possible for each observation, $y \in Y$. Let $\Theta(y)$ denote this, which is obtained as

$$\Theta(y) = \{\theta \in \Theta \mid y \in Y(\theta)\}. \tag{9.23}$$

The optimal strategy, π^*, is defined by setting

$$\pi^*(y) = \operatorname*{argmin}_{u \in U} \Big\{ \max_{\theta \in \Theta(y)} \Big\{ L(u, \theta) \Big\} \Big\}, \tag{9.24}$$

for each $y \in Y$. Compare this to (9.14), in which the maximum was taken over all Θ. The advantage of having the observation, y, is that the set is restricted to $\Theta(y) \subseteq \Theta$.

Under the probabilistic model, an operation analogous to (9.23) must be performed. This involves computing $P(\theta|y)$ from $P(y|\theta)$ to determine the information that y contains regarding θ. Using Bayes' rule, (9.9), with marginalization on the denominator, the result is

$$P(\theta|y) = \frac{P(y|\theta)P(\theta)}{\sum_{\theta \in \Theta} P(y|\theta)P(\theta)}. \tag{9.25}$$

To see the connection between the nondeterministic and probabilistic cases, define a probability distribution, $P(y|\theta)$, that is nonzero only if $y \in Y(\theta)$ and use a uniform distribution for $P(\theta)$. In this case, (9.25) assigns nonzero probability to precisely the elements of $\Theta(y)$ as given in (9.23). Thus, (9.25) is just the

probabilistic version of (9.23). The optimal strategy, π^*, is specified for each $y \in Y$ as

$$\pi^*(y) = \operatorname*{argmin}_{u \in U} \left\{ E_\theta \Big[L(u, \theta) \;\Big|\; y \Big] \right\} = \operatorname*{argmin}_{u \in U} \left\{ \sum_{\theta \in \Theta} L(u, \theta) P(\theta | y) \right\}. \tag{9.26}$$

This differs from (9.15) and (9.16) by replacing $P(\theta)$ with $P(\theta|y)$. For each u, the expectation in (9.26) is called the *conditional Bayes' risk*. The optimal strategy, π^*, always selects the strategy that minimizes this risk. Note that $P(\theta|y)$ in (9.26) can be expressed using (9.25), for which the denominator (9.26) represents $P(y)$ and does not depend on u; therefore, it does not affect the optimization. Due to this, $P(y|\theta)P(\theta)$ can be used in the place of $P(\theta|y)$ in (9.26), and the same π^* will be obtained. If the spaces are continuous, then probability densities are used in the place of all probability distributions, and the method otherwise remains the same.

Nature acts twice A convenient, alternative formulation can be given by allowing nature to act twice:

1. First, a nature action, $\theta \in \Theta$, is chosen but is unknown to the robot.

2. Following this, a *nature observation action* is chosen to interfere with the robot's ability to sense θ.

Let ψ denote a *nature observation action*, which is chosen from a *nature observation action space*, $\Psi(\theta)$. A *sensor mapping*, h, can now be defined that yields $y = h(\theta, \psi)$ for each $\theta \in \Theta$ and $\psi \in \Psi(\theta)$. Thus, for each of the two kinds of nature actions, $\theta \in \Theta$ and $\psi \in \Psi$, an observation, $y = h(\theta, \psi)$, is given. This yields an alternative way to express Formulation 9.5:

Formulation 9.6 (Nature Interferes with the Observation)

1. A nonempty, finite set U called the *action space*.

2. A nonempty, finite set Θ called the *nature action space*.

3. A nonempty, finite set Y called the *observation space*.

4. For each $\theta \in \Theta$, a nonempty set $\Psi(\theta)$ called the *nature observation action space*.

5. A sensor mapping $h : \Theta \times \Psi \to Y$.

6. A function $L : U \times \Theta \to \mathbb{R} \cup \{\infty\}$ called the *cost function*.

This nicely unifies the nondeterministic and probabilistic models with a single function h. To express a nondeterministic model, it is assumed that any $\psi \in \Psi(\theta)$ is possible. Using h,

$$\Theta(y) = \{\theta \in \Theta \mid \exists \psi \in \Psi(\theta) \text{ such that } y = h(\theta, \psi)\}. \tag{9.27}$$

For a probabilistic model, a distribution $P(\psi|\theta)$ is specified (often, this may reduce to $P(\psi)$). Suppose that when the domain of h is restricted to some $\theta \in \Theta$, then it forms an injective mapping from Ψ to Y. In other words, every nature observation action leads to a unique observation, assuming θ is fixed. Using $P(\psi)$ and h, $P(y|\theta)$ is derived as

$$P(y|\theta) = \begin{cases} P(\psi|\theta) & \text{for the unique } \psi \text{ such that } y = h(\theta, \psi). \\ 0 & \text{if no such } \psi \text{ exists.} \end{cases} \tag{9.28}$$

If the injective assumption is lifted, then $P(\psi|\theta)$ is replaced by a sum over all ψ for which $y = h(\theta, \psi)$. In Formulation 9.6, the only difference between the nondeterministic and probabilistic models is the characterization of ψ, which represents a kind of measurement interference. A strategy still takes the form $\pi : \Theta \rightarrow U$. A hybrid model is even possible in which one nature action is modeled nondeterministically and the other probabilistically.

Receiving multiple observations Another extension of Formulation 9.5 is to allow multiple observations, y_1, y_2, ..., y_n, before making a decision. Each y_i is assumed to belong to an observation space, Y_i. A strategy, π, now depends on all observations:

$$\pi : Y_1 \times Y_2 \times \cdots \times Y_n \rightarrow U. \tag{9.29}$$

Under the nondeterministic model, $Y_i(\theta)$ is specified for each i and $\theta \in \Theta$. The set $\Theta(y)$ is replaced by

$$\Theta(y_1) \cap \Theta(y_2) \cap \cdots \cap \Theta(y_n) \tag{9.30}$$

in (9.24) to obtain the optimal action, $\pi^*(y_1, \ldots, y_n)$.

Under the probabilistic model, $P(y_i|\theta)$ is specified instead. It is often assumed that the observations are conditionally independent given θ. This means for any y_i, θ, and y_j such that $i \neq j$, $P(y_i|\theta, y_j) = P(y_i|\theta)$. The condition $P(\theta|y)$ in (9.26) is replaced by $P(\theta|y_1, \ldots, y_n)$. Applying Bayes' rule, and using the conditional independence of the y_i's given θ, yields

$$P(\theta|y_1, \ldots, y_n) = \frac{P(y_1|\theta)P(y_2|\theta) \cdots P(y_n|\theta)P(\theta)}{P(y_1, \ldots, y_n)}. \tag{9.31}$$

The denominator can be treated as a constant factor that does not affect the optimization. Therefore, it does not need to be explicitly computed unless the optimal expected cost is needed in addition to the optimal action.

Conditional independence allows a dramatic simplification that avoids the full specification of $P(y|\theta)$. Sometimes the conditional independence assumption is used when it is incorrect, just to exploit this simplification. Therefore, a method that uses conditional independence of observations is often called *naive Bayes*.

9.2.4 Examples of Optimal Decision Making

The framework presented so far characterizes *statistical decision theory*, which covers a broad range of applications and research issues. Virtually any context in which a decision must be made automatically, by a machine or a person following specified rules, is a candidate for using these concepts. In Chapters 10 through 12, this decision problem will be repeatedly embedded into complicated planning problems. Planning will be viewed as a sequential decision-making process that iteratively modifies states in a state space. Most often, each decision step will be simpler than what usually arises in common applications of decision theory. This is because planning problems are complicated by many other factors. If the decision step in a particular application is already too hard to solve, then an extension to planning appears hopeless.

It is nevertheless important to recognize the challenges in general that arise when modeling and solving decision problems under the framework of this section. Some examples are presented here to help illustrate its enormous power and scope.

9.2.4.1 Pattern classification

An active field over the past several decades in computer vision and machine learning has been *pattern classification* [271, 295, 711]. The general problem involves using a set of data to perform classifications. For example, in computer vision, the data correspond to information extracted from an image. These indicate observed features of an object that are used by a vision system to try to classify the object (e.g., "I am looking at a bowl of Vietnamese noodle soup").

The presentation here represents a highly idealized version of pattern classification. We will assume that all of the appropriate model details, including the required probability distributions, are available. In some contexts, these can be obtained by gathering statistics over large data sets. In many applications, however, obtaining such data is expensive or inaccessible, and classification techniques must be developed in lieu of good information. Some problems are even *unsupervised*, which means that the set of possible classes must also be discovered automatically. Due to issues such as these, pattern classification remains a challenging research field.

The general model is that nature first determines the class, then observations are obtained regarding the class, and finally the robot action attempts to guess the correct class based on the observations. The problem fits under Formulation 9.5. Let Θ denote a finite set of *classes*. Since the robot must guess the class, $U = \Theta$. A simple cost function is defined to measure the mismatch between u and θ:

$$L(u, \theta) = \begin{cases} 0 & \text{if } u = \theta \text{ (correct classification} \\ 1 & \text{if } u \neq \theta \text{ (incorrect classification) .} \end{cases} \tag{9.32}$$

The nondeterministic model yields a cost of 1 if it is *possible* that a classification error can be made using action u. Under the probabilistic model, the expectation

of (9.32) gives the probability that a classification error will be made given an action u.

The next part of the formulation considers information that is used to make the classification decision. Let Y denote a *feature space*, in which each $y \in Y$ is called a *feature* or *feature vector* (often $y \in \mathbb{R}^n$). The feature in this context is just an observation, as given in Formulation 9.5. The best *classifier* or *classification rule* is a strategy $\pi : Y \rightarrow U$ that provides the smallest classification error in the worst case or expected case, depending on the model.

A Bayesian classifier The probabilistic approach is most common in pattern classification. This results in a *Bayesian classifier*. Here it is assumed that $P(y|\theta)$ and $P(\theta)$ are given. The distribution of features for a given class is indicated by $P(y|\theta)$. The overall frequency of class occurrences is given by $P(\theta)$. If large, pre-classified data sets are available, then these distributions can be reliably learned. The feature space is often continuous, which results in a density $p(y|\theta)$, even though $P(\theta)$ remains a discrete probability distribution. An optimal classifier, π^*, is designed according to (9.26). It performs classification by receiving a feature vector, y, and then declaring that the class is $u = \pi^*(y)$. The expected cost using (9.32) is the probability of error.

Example 9.11 (Optical Character Recognition) An example of classification is given by a simplified *optical character recognition* (OCR) problem. Suppose that a camera creates a digital image of a page of text. Segmentation is first performed to determine the location of each letter. Following this, the individual letters must be classified correctly. Let $\Theta = \{A, B, C, D, E, F, G, H\}$, which would ordinarily include all of the letters of the alphabet.

Suppose that there are three different image processing algorithms:

Shape extractor: This returns $s = 0$ if the letter is composed of straight edges only, and $s = 1$ if it contains at least one curve.

End counter: This returns e, the number of segment ends. For example, O has none and X has four.

Hole counter: This returns h, the number of holes enclosed by the character. For example, X has none and O has one.

The feature vector is $y = (s, e, h)$. The values that should be reported under ideal conditions are shown in Figure 9.1. These indicate $\Theta(s)$, $\Theta(e)$, and $\Theta(h)$. The intersection of these yields $\Theta(y)$ for any combination of s, e, and h.

Imagine doing classification under the nondeterministic model, with the assumption that the features always provide correct information. For $y = (0, 2, 1)$, the only possible letter is A. For $y = (1, 0, 2)$, the only letter is B. If each (s, e, h) is consistent with only one or no letters, then a perfect classifier can be constructed. Unfortunately, $(0, 3, 0)$ is consistent with both E and F. In the worst case, the cost of using (9.32) is 1.

Shape	0	A E F H
	1	B C D G
Ends	0	B D
	1	
	2	A C G
	3	F E
	4	H
Holes	0	C E F G H
	1	A D
	2	B

Figure 9.1: A mapping from letters to feature values.

One way to fix this is to introduce a new feature. Suppose that an image processing algorithm is used to detect corners. These are places at which two segments meet at a right (90 degrees) angle. Let c denote the number of corners, and let the new feature vector be $y = (s, e, h, c)$. The new algorithm nicely distinguishes E from F, for which $c = 2$ and $c = 1$, respectively. Now all letters can be correctly classified without errors.

Of course, in practice, the image processing algorithms occasionally make mistakes. A Bayesian classifier can be designed to maximize the probability of success. Assume conditional independence of the observations, which means that the classifier can be considered *naive*. Suppose that the four image processing algorithms are run over a training data set and the results are recorded. In each case, the correct classification is determined by hand to obtain probabilities $P(s|\theta)$, $P(e|\theta)$, $P(h|\theta)$, and $P(c|\theta)$. For example, suppose that the hole counter receives the letter A as input. After running the algorithm over many occurrences of A in text, it may be determined that $P(h = 1|\ \theta = A) = 0.9$, which is the correct answer. With smaller probabilities, perhaps $P(h = 0|\ \theta = A) = 0.09$ and $P(h = 2|\ \theta = A) = 0.01$. Assuming that the output of each image processing algorithm is independent given the input letter, a joint probability can be assigned as

$$P(y|\theta) = P(s, e, h, c|\ \theta) = P(s|\theta)P(e|\theta)P(h|\theta)P(c|\theta). \tag{9.33}$$

The value of the prior $P(\theta)$ can be obtained by running the classifier over large amounts of hand-classified text and recording the relative numbers of occurrences of each letter. It is interesting to note that some context-specific information can be incorporated. If the text is known to be written in Spanish, then $P(\theta)$ should be different than from text written in English. Tailoring $P(\theta)$ to the type of text that will appear improves the performance of the resulting classifier.

The classifier makes its decisions by choosing the action that minimizes the probability of error. This error is proportional to

$$\sum_{\theta \in \Theta} P(s|\theta)P(e|\theta)P(h|\theta)P(c|\theta)P(\theta), \tag{9.34}$$

by neglecting the constant $P(y)$ in the denominator of Bayes' rule in (9.26). ■

9.2.4.2 Parameter estimation

Another important application of the decision-making framework of this section is *parameter estimation* [89, 268]. In this case, nature selects a *parameter*, $\theta \in \Theta$, and Θ represents a *parameter space*. Through one or more independent trials, some observations are obtained. Each observation should ideally be a direct measurement of Θ, but imperfections in the measurement process distort the observation. Usually, $\Theta \subseteq Y$, and in many cases, $Y = \Theta$. The robot action is to guess the parameter that was chosen by nature. Hence, $U = \Theta$. In most applications, all of the spaces are continuous subsets of $\mathbb{R}^n$. The cost function is designed to increase as the error, $\|u - \theta\|$, becomes larger.

Example 9.12 (Parameter Estimation) Suppose that $U = Y = \Theta = \mathbb{R}$. Nature therefore chooses a real-valued parameter, which is estimated. The cost of making a mistake is

$$L(u, \theta) = (u - \theta)^2. \tag{9.35}$$

Suppose that a Bayesian approach is taken. The prior probability density $p(\theta)$ is given as uniform over an interval $[a, b] \subset \mathbb{R}$. An observation is received, but it is noisy. The noise can be modeled as a second action of nature, as described in Section 9.2.3. This leads to a density $p(y|\theta)$. Suppose that the noise is modeled with a Gaussian, which results in

$$p(y|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-(y-\theta)^2/2\sigma^2}, \tag{9.36}$$

in which the mean is θ and the standard deviation is σ.

The optimal parameter estimate based on y is obtained by selecting $u \in \mathbb{R}$ to minimize

$$\int_{-\infty}^{\infty} L(u, \theta) p(\theta|y) d\theta, \tag{9.37}$$

in which

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}, \tag{9.38}$$

by Bayes' rule. The term $p(y)$ does not depend on θ, and it can therefore be ignored in the optimization. Using the prior density, $p(\theta) = 0$ outside of $[a, b]$; hence, the domain of integration can be restricted to $[a, b]$. The value of $p(\theta) = 1/(b - a)$ is also a constant that can be ignored in the optimization. Using (9.36), this means that u is selected to optimize

$$\int_{a}^{b} L(u, \theta) p(y|\theta) d\theta, \tag{9.39}$$

which can be expressed in terms of the standard error function, $\mathrm{erf}(x)$ (the integral from 0 to a constant, of a Gaussian density over an interval).

If a sequence, $y_1, \ldots, y_k$, of independent observations is obtained, then (9.39) is replaced by

$$\int_a^b L(u, \theta) p(y_1|\theta) \cdots p(y_k|\theta) d\theta. \tag{9.40}$$

■

9.3 Two-Player Zero-Sum Games

Section 9.2 involved one real decision maker (DM), the robot, playing against a fictitious DM called nature. Now suppose that the second DM is a clever opponent that makes decisions in the same way that the robot would. This leads to a symmetric situation in which two decision makers simultaneously make a decision, without knowing how the other will act. It is assumed in this section that the DMs have diametrically opposing interests. They are two players engaged in a game in which a loss for one player is a gain for the other, and vice versa. This results in the most basic form of *game theory*, which is referred to as a *zero-sum game*.

9.3.1 Game Formulation

Suppose there are two *players*, P_1 and P_2, that each have to make a decision. Each has a finite set of actions, U and V, respectively. The set V can be viewed as the "replacement" of Θ from Formulation 9.3 by a set of actions chosen by a true opponent. Each player has a cost function, which is denoted as $L_i : U \times V \rightarrow \mathbb{R}$ for $i = 1, 2$. An important constraint for zero-sum games is

$$L_1(u, v) = -L_2(u, v), \tag{9.41}$$

which means that a cost for one player is a reward for the other. This is the basis of the term *zero sum*, which means that the two costs can be added to obtain zero. In zero-sum games the interests of the players are completely opposed. In Section 9.4 this constraint will be lifted to obtain more general games.

In light of (9.41) it is pointless to represent two cost functions. Instead, the superscript will be dropped, and L will refer to the cost, L_1, of P_1. The goal of P_1 is to minimize L. Due to (9.41), the goal of P_2 is to maximize L. Thus, L can be considered as a *reward* for P_2, but a *cost* for P_1.

A formulation can now be given:

Formulation 9.7 (A Zero-Sum Game)

1. Two players, P_1 and P_2.

2. A nonempty, finite set U called the *action space for* P_1. For convenience in describing examples, assume that U is a set of consecutive integers from 1 to $|U|$. Each $u \in U$ is referred to as an *action of* P_1.

3. A nonempty, finite set V called the *action space for* P_2. Assume that V is a set of consecutive integers from 1 to $|V|$. Each $v \in V$ is referred to as an *action of* P_2.

4. A function $L : U \times V \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$ called the *cost function* for P_1. This also serves as a *reward function* for P_2 because of (9.41).

Before discussing what it means to solve a zero-sum game, some additional assumptions are needed. Assume that the players know each other's cost functions. This implies that the motivation of the opponent is completely understood. The other assumption is that the players are *rational*, which means that they will try to obtain the best cost whenever possible. P_1 will not choose an action that leads to higher cost when a lower cost action is available. Likewise, P_2 will not choose an action that leads to lower cost. Finally, it is assumed that both players make their decisions simultaneously. There is no information regarding the decision of P_1 that can be exploited by P_2, and vice versa.

Formulation 9.7 is often referred to as a *matrix game* because L can be expressed with a cost matrix, as was done in Section 9.2. Here the matrix indicates costs for P_1 and P_2, instead of the robot and nature. All of the required information from Formulation 9.7 is specified by a single matrix; therefore, it is a convenient form for expressing zero-sum games.

Example 9.13 (Matrix Representation of a Zero-Sum Game) Suppose that U, the action set for P_1, contains three actions and V contains four actions. There should be $3 \times 4 = 12$ values in the specification of the cost function, L. This can be expressed as a cost matrix,

$$\begin{array}{cc} & V \\ U & \begin{array}{|c|c|c|c|} \hline 1 & 3 & 3 & 2 \\ \hline 0 & -1 & 2 & 1 \\ \hline -2 & 2 & 0 & 1 \\ \hline \end{array} \end{array}, \tag{9.42}$$

in which each row corresponds to some $u \in U$, and each column corresponds to some $v \in V$. Each entry yields $L(u, v)$, which is the cost for P_1. This representation is similar to that shown in Example 9.8, except that the nature action space, Θ, is replaced by V. The cost for P_2 is $-L(u, v)$. ■

9.3.2 Deterministic Strategies

What constitutes a good solution to Formulation 9.7? Consider the game from the perspective of P_1. It seems reasonable to apply worst-case analysis when trying

to account for the action that will be taken by P_2. This results in a choice that is equivalent to assuming that P_2 is nature acting under the nondeterministic model, as considered in Section 9.2.2. For a matrix game, this is computed by first determining the maximum cost over each row. Selecting the action that produces the minimum among these represents the lowest cost that P_1 can guarantee for itself. Let this selection be referred to as a *security strategy* for P_1.

For the matrix game in (9.42), the security strategy is illustrated as

$$\begin{array}{cc} & V \\ U & \begin{array}{|cccc|c|} \hline 1 & 3 & 3 & 2 & \rightarrow 3 \\ \hline 0 & -1 & 2 & 1 & \rightarrow \mathbf{2} \\ \hline -2 & 2 & 0 & 1 & \rightarrow \mathbf{2} \\ \hline \end{array} \end{array}, \tag{9.43}$$

in which $u = 2$ and $u = 3$ are the best actions. Each yields a cost no worse than 2, regardless of the action chosen by P_2.

This can be formalized using the existing notation. A security strategy, u^*, for P_1 is defined in general as

$$u^* = \operatorname*{argmin}_{u \in U} \Big\{ \max_{v \in V} \Big\{ L(u, v) \Big\} \Big\}. \tag{9.44}$$

There may be multiple security strategies that satisfy the argmin; however, this does not cause trouble, as will be explained shortly. Let the resulting worst-case cost be denoted by $\overline{L}^*$, and let it be called the *upper value* of the game. This is defined as

$$\overline{L}^* = \max_{v \in V} \Big\{ L(u^*, v) \Big\}. \tag{9.45}$$

Now swap roles, and consider the game from the perspective of P_2, which would like to maximize L. It can also use worst-case analysis, which means that it would like to select an action that guarantees a high cost, in spite of the action of P_1 to potentially reduce it. A security strategy, v^*, for P_2 is defined as

$$v^* = \operatorname*{argmax}_{v \in V} \Big\{ \min_{u \in U} \Big\{ L(u, v) \Big\} \Big\}. \tag{9.46}$$

Note the symmetry with respect to (9.44). There may be multiple security strategies for P_2. A security strategy v^* is just an "upside-down" version of the worst-case analysis applied in Section 9.2.2. The *lower value*, $\underline{L}^*$, is defined as

$$\underline{L}^* = \min_{u \in U} \Big\{ L(u, v^*) \Big\}. \tag{9.47}$$

Returning to the matrix game in (9.42), the last column is selected by applying (9.46):

$$\begin{array}{cc} & V \\ U & \begin{array}{|cccc|} \hline 1 & 3 & 3 & 2 \\ 0 & -1 & 2 & 1 \\ -2 & 2 & 0 & 1 \\ \hline \downarrow & \downarrow & \downarrow & \downarrow \\ -2 & -1 & 0 & \mathbf{1} \\ \hline \end{array} \end{array}. \tag{9.48}$$

An interesting relationship between the upper and lower values is that $\underline{L}^* \leq \overline{L}^*$ for any game using Formulation 9.7. This is shown by observing that

$$\underline{L}^* = \min_{u \in U} \Big\{ L(u, v^*) \Big\} \leq L(u^*, v^*) \leq \max_{v \in V} \Big\{ L(u^*, v) \Big\} = \overline{L}^*, \tag{9.49}$$

in which $L(u^*, v^*)$ is the cost received when the players apply their respective security strategies. If the game is played by rational DMs, then the resulting cost always lies between $\underline{L}^*$ and $\overline{L}^*$.

Regret Suppose that the players apply security strategies, $u^* = 2$ and $v^* = 4$. This results in a cost of $L(2, 4) = 1$. How do the players feel after the outcome? P_1 may feel satisfied because given that P_2 selected $v^* = 4$, it received the lowest cost possible. On the other hand, P_2 may regret its decision in light of the action chosen by P_1. If it had known that $u = 2$ would be chosen, then it could have picked $v = 2$ to receive cost $L(2, 2) = 2$, which is better than $L(2, 4) = 1$. If the game were to be repeated, then P_2 would want to change its strategy in hopes of tricking P_1 to obtain a higher reward.

Is there a way to keep both players satisfied? Any time there is a gap between $\underline{L}^*$ and $\overline{L}^*$, there is regret for one or both players. If r_1 and r_2 denote the amount of regret experienced by P_1 and P_2, respectively, then the total regret is

$$r_1 + r_2 = \overline{L}^* - \underline{L}^*. \tag{9.50}$$

Thus, the only way to satisfy both players is to obtain upper and lower values such that $\underline{L}^* = \overline{L}^*$. These are properties of the game, however, and they are not up to the players to decide. For some games, the values are equal, but for many $\underline{L}^* < \overline{L}^*$. Fortunately, by using randomized strategies, the upper and lower values always coincide; this is covered in Section 9.3.3.

Saddle points If $\underline{L}^* = \overline{L}^*$, the security strategies are called a *saddle point*, and $L^* = \underline{L}^* = \overline{L}^*$ is called the *value* of the game. If this occurs, the order of the max and min can be swapped without changing the value:

$$L^* = \min_{u \in U} \Big\{ \max_{v \in V} \Big\{ L(u, v) \Big\} \Big\} = \max_{v \in V} \Big\{ \min_{u \in U} \Big\{ L(u, v) \Big\} \Big\}. \tag{9.51}$$

A saddle point is sometimes referred to as an *equilibrium* because the players have no incentive to change their choices (because there is no regret). A saddle point is defined as any $u^* \in U$ and $v^* \in V$ such that

$$L(u^*, v) \leq L(u^*, v^*) \leq L(u, v^*) \tag{9.52}$$

for all $u \in U$ and $v \in V$. Note that $L^* = L(u^*, v^*)$. When looking at a matrix game, a saddle point is found by finding the simple pattern shown in Figure 9.2.

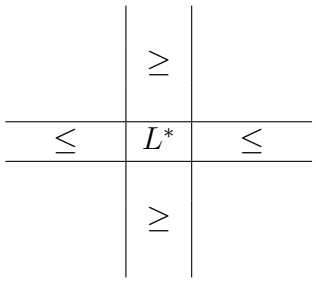

Figure 9.2: A saddle point can be detected in a matrix by finding a value L^* that is lowest among all elements in its column and greatest among all elements in its row.

Example 9.14 (A Deterministic Saddle Point) Here is a matrix game that has a saddle point:

$$\begin{array}{cc} & V \\ U & \begin{array}{|c|c|c|} \hline 3 & 3 & 5 \\ \hline 1 & -1 & 7 \\ \hline 0 & -2 & 4 \\ \hline \end{array} \end{array} . \tag{9.53}$$

By applying (9.52) (or using Figure 9.2), the saddle point is obtained when $u = 3$ and $v = 3$. The result is that $L^* = 4$. In this case, neither player has regret after the game is finished. P_1 is satisfied because 4 is the lowest cost it could have received, given that P_2 chose the third column. Likewise, 4 is the highest cost that P_2 could have received, given that P_1 chose the bottom row. ■

What if there are multiple saddle points in the same game? This may appear to be a problem because the players have no way to coordinate their decisions. What if P_1 tries to achieve one saddle point while P_2 tries to achieve another? It turns out that if there is more than one saddle point, then there must at least be four, as shown in Figure 9.3. As soon as we try to make two "+" patterns like the one shown in Figure 9.2, they intersect, and four saddle points are created. Similar behavior occurs as more saddle points are added.

Example 9.15 (Multiple Saddle Points) This game has multiple saddle points and follows the pattern in Figure 9.3:

$$\begin{array}{cc} & V \\ U & \begin{array}{|c|c|c|c|c|} \hline 4 & 3 & 5 & 1 & 2 \\ \hline -1 & 0 & -2 & 0 & -1 \\ \hline -4 & 1 & 4 & 3 & 5 \\ \hline -3 & 0 & -1 & 0 & -2 \\ \hline 3 & 2 & -7 & 3 & 8 \\ \hline \end{array} \end{array} . \tag{9.54}$$

Let (i, j) denote the pair of choices for P_1 and P_2, respectively. Both $(2, 2)$ and $(4, 4)$ are saddle points with value $V = 0$. What if P_1 chooses $u = 2$ and P_2 chooses

	$\geq$		$\geq$	
$\leq$	L^*	$\leq$	L^*	$\leq$
	$\geq$		$\geq$	
$\leq$	L^*	$\leq$	L^*	$\leq$
	$\geq$		$\geq$	

Figure 9.3: A matrix could have more than one saddle point, which may seem to lead to a coordination problem between the players. Fortunately, there is no problem, because the same value will be received regardless of which saddle point is selected by each player.

$v = 4$? This is not a problem because $(2, 4)$ is also a saddle point. Likewise, $(4, 2)$ is another saddle point. In general, no problems are caused by the existence of multiple saddle points because the resulting cost is independent of which saddle point is attempted by each player. ■

9.3.3 Randomized Strategies

The fact that some zero-sum games do not have a saddle point is disappointing because regret is unavoidable in these cases. Suppose we slightly change the rules. Assume that the same game is repeatedly played by P_1 and P_2 over numerous trials. If they use a deterministic strategy, they will choose the same actions every time, resulting in the same costs. They may instead switch between alternative security strategies, which causes fluctuations in the costs. What happens if they each implement a randomized strategy? Using the idea from Section 9.1.3, each strategy is specified as a probability distribution over the actions. In the limit, as the number of times the game is played tends to infinity, an expected cost is obtained. One of the most famous results in game theory is that on the space of randomized strategies, a saddle point always exists for any zero-sum matrix game; however, expected costs must be used. Thus, if randomization is used, there will be no regrets. In an individual trial, regret may be possible; however, as the costs are averaged over all trials, both players will be satisfied.

9.3.3.1 Extending the formulation

Since a game under Formulation 9.7 can be nicely expressed as a matrix, it is tempting to use linear algebra to conveniently express expected costs. Let $|U| = m$ and $|V| = n$. As in Section 9.1.3, a randomized strategy for P_1 can be represented as an m-dimensional vector,

$$w = [w_1 \;\; w_2 \;\; \ldots \;\; w_m]. \tag{9.55}$$

The probability axioms of Section 9.1.2 must be satisfied: 1) $w_i \geq 0$ for all $i \in \{1, \ldots, m\}$, and 2) $w_1 + \cdots + w_m = 1$. If w is considered as a point in $\mathbb{R}^m$, then the two constraints imply that it must lie on an $(m-1)$-dimensional simplex (recall Section 6.3.1). If $m = 3$, this means that w lies in a triangular subset of $\mathbb{R}^3$. Similarly, let z represent a randomized strategy for P_2 as an n-dimensional vector,

$$z = [z_1 \;\; z_2 \;\; \ldots \;\; z_n]^T, \tag{9.56}$$

that also satisfies the probability axioms. In (9.56), ^{T} denotes *transpose*, which yields a column vector that satisfies the dimensional constraints required for an upcoming matrix multiplication.

Let $\bar{L}(w, z)$ denote the expected cost that will be received if P_1 plays w and P_2 plays z. This can be computed as

$$\bar{L}(w, z) = \sum_{i=1}^{m} \sum_{j=1}^{n} L(i, j) w_i z_j. \tag{9.57}$$

Note that the cost, $L(i, j)$, makes use of the assumption in Formulation 9.7 that the actions are consecutive integers. The expected cost can be alternatively expressed using the cost matrix, A. In this case

$$\bar{L}(w, z) = wAz, \tag{9.58}$$

in which the product wAz yields a scalar value that is precisely (9.57). To see this, first consider the product Az. This yields an m-dimensional vector in which the ith element is the expected cost that P_1 would receive if it tries $u = i$. Thus, it appears that P_1 views P_2 as a nature player under the probabilistic model. Once w and Az are multiplied, a scalar value is obtained, which averages the costs in the vector Az according the probabilities of w.

Let W and Z denote the set of all randomized strategies for P_1 and P_2, respectively. These spaces include strategies that are equivalent to the deterministic strategies considered in Section 9.3.2 by assigning probability one to a single action. Thus, W and Z can be considered as expansions of the set of possible strategies in comparison to what was available in the deterministic setting. Using W and Z, *randomized security strategies* for P_1 and P_2 are defined as

$$w^* = \operatorname*{argmin}_{w \in W} \left\{ \max_{z \in Z} \left\{ \bar{L}(w, z) \right\} \right\} \tag{9.59}$$

and

$$z^* = \operatorname*{argmax}_{z \in Z} \left\{ \min_{w \in W} \left\{ \bar{L}(w, z) \right\} \right\}, \tag{9.60}$$

respectively. These should be compared to (9.44) and (9.46). The differences are that the space of strategies has been expanded, and expected cost is now used.

The *randomized upper value* is defined as

$$\overline{\mathcal{L}}^* = \max_{z \in Z} \left\{ \bar{L}(w^*, z) \right\}, \tag{9.61}$$

and the *randomized lower value* is

$$\underline{\mathcal{L}}^* = \min_{w \in W} \Big\{ \bar{L}(w, z^*) \Big\}. \tag{9.62}$$

Since W and Z include the deterministic security strategies, $\overline{\mathcal{L}}^* \leq \overline{L}^*$ and $\underline{\mathcal{L}}^* \geq \underline{L}^*$. These inequalities imply that the randomized security strategies may have some hope in closing the gap between the two values in general.

The most fundamental result in zero-sum game theory was shown by von Neumann [956, 957], and it states that $\underline{\mathcal{L}}^* = \overline{\mathcal{L}}^*$ for any game in Formulation 9.7. This yields the *randomized value* $\mathcal{L}^* = \underline{\mathcal{L}}^* = \overline{\mathcal{L}}^*$ for the game. This means that there will never be expected regret if the players stay with their security strategies. If the players apply their randomized security strategies, then a *randomized saddle point* is obtained. This saddle point cannot be seen as a simple pattern in the matrix A because it instead exists over W and Z.

The guaranteed existence of a randomized saddle point is an important result because it demonstrates the value of randomization when making decisions against an intelligent opponent. In Example 9.7, it was intuitively argued that randomization seems to help when playing against an intelligent adversary. When playing the game repeatedly with a deterministic strategy, the other player could learn the strategy and win every time. Once a randomized strategy is used, the players will not experience regret.

9.3.3.2 Computation of randomized saddle points

So far it has been established that a randomized saddle point always exists, but how can one be found? Two key observations enable a combinatorial solution to the problem:

1. The security strategy for each player can be found by considering only deterministic strategies for the opposing player.

2. If the strategy for the other player is fixed, then the expected cost is a linear function of the undetermined probabilities.

First consider the problem of determining the security strategy for P_1. The first observation means that (9.59) does not need to consider randomized strategies for P_2. Inside of the argmin, w is fixed. What randomized strategy, $z \in Z$, maximizes $\bar{L}(w, z) = wAz$? If w is fixed, then wA can be treated as a constant n-dimensional vector, s. This means $\bar{L}(w, z) = s \cdot z$, in which $\cdot$ is the inner (dot) product. Now the task is to select z to maximize $s \cdot z$. This involves selecting the largest element of s; suppose this is s_i. The maximum cost over all $z \in Z$ is obtained by placing all of the probability mass at action i. Thus, the strategy $z_i = 1$ and $z_j = 0$ for $i \neq j$ gives the highest cost, and it is deterministic.

Using the first observation, for each $w \in W$, only n possible responses by P_2 need to be considered. These are the n deterministic strategies, each of which assigns $z_i = 1$ for a unique $i \in \{1, \ldots, n\}$.

Now consider the second observation. The expected cost, $\bar{L}(w, z) = wAz$, is a linear function of w, if z is fixed. Since z only needs to be fixed at n different values due to the first observation, w is selected at the point at which the smallest maximum value among the n linear functions occurs. This is the minimum value of the *upper envelope* of the collection of linear functions. Such envelopes were mentioned in Section 6.5.2. Example 9.16 will illustrate this. The domain for this optimization can conveniently be set as a triangle in $\mathbb{R}^{m-1}$. Even though $W \subset \mathbb{R}^m$, the last coordinate, w_m, is not needed because it is always $w_m = 1 - (w_1 + \cdots + w_{m-1})$. The resulting optimization falls under *linear programming*, for which many combinatorial algorithms exist [259, 264, 664, 731].

In the explanation above, there is nothing particular to P_1 when trying to find its security strategy. The same method can be applied to determine the security strategy for P_2; however, every minimization is replaced by a maximization, and vice versa. In summary, the min in (9.60) needs only to consider the deterministic strategies in W. If w becomes fixed, then $\bar{L}(w, z) = wAz$ is once again a linear function, but this time it is linear in z. The best randomized action is chosen by finding the point $z \in Z$ that gives the highest minimum value among m linear functions. This is the minimum value of the *lower envelope* of the collection of linear functions. The optimization occurs over $\mathbb{R}^{n-1}$ because the last coordinate, z_n, is obtained directly from $z_n = 1 - (z_1 + \cdots + z_{n-1})$.

This computation method is best understood through an example.

Example 9.16 (Computing a Randomized Saddle Point) The simplest case is when both players have only two actions. Let the cost matrix be defined as

		V	
U		3	0
		-1	1

. (9.63)

Consider computing the security strategy for P_1. Note that W and Z are only one-dimensional subsets of $\mathbb{R}^2$. A randomized strategy for P_1 is $w = [w_1 \;\; w_2]$, with $w_1 \geq 0$, $w_2 \geq 0$, and $w_1 + w_2 = 1$. Therefore, the domain over which the optimization is performed is $w_1 \in [0, 1]$ because w_2 can always be derived as $w_2 = 1 - w_1$. Using the first observation above, only the two deterministic strategies for P_2 need to be considered. When considered as linear functions of w, these are

$$(3)w_1 + (-1)(1 - w_1) = 4w_1 - 1 \tag{9.64}$$

for $z_1 = 1$ and

$$(0)w_1 + (1)(1 - w_1) = 1 - w_1 \tag{9.65}$$

for $z_2 = 1$. The lines are plotted in Figure 9.4a. The security strategy is determined by the minimum point along the upper envelope shown in the figure. This is indicated by the thickened line, and it is always a piecewise-linear function in general. The lowest point occurs at $w_1 = 2/5$, and the resulting value is $\mathcal{L}^* = 3/5$. Therefore, $w^* = [2/5 \;\; 3/5]$.

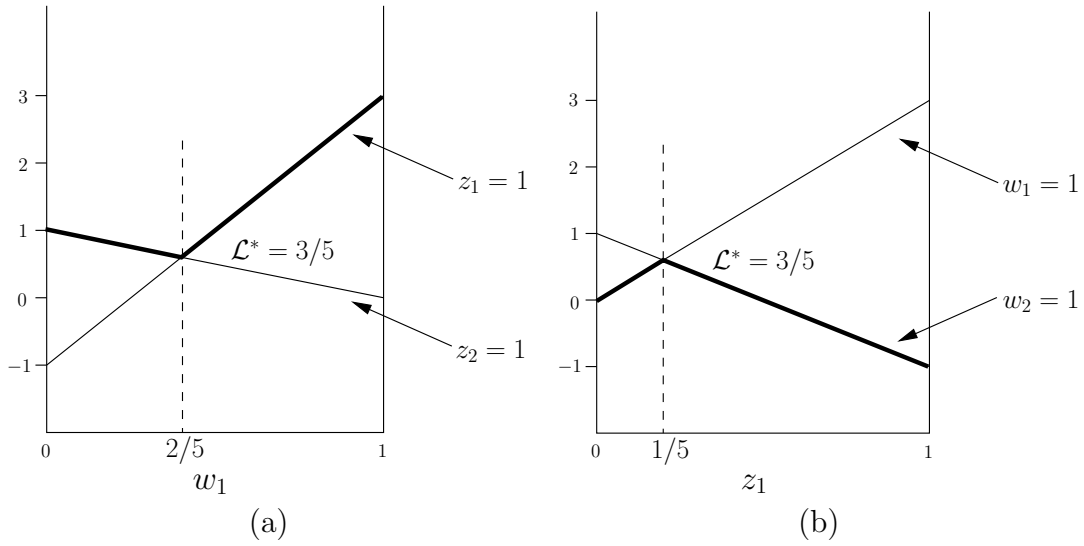

Figure 9.4: (a) Computing the randomized security strategy, w^*, for P_1. (b) Computing the randomized security strategy, z^*, for P_2.

A similar procedure can be used to obtain z^*. The lines that correspond to the deterministic strategies of P_1 are shown in Figure 9.4b. The security strategy is obtained by finding the maximum value along the lower envelope of the lines, which is shown as the thickened line in the figure. This results in $z^* = [1/5 \;\; 4/5]^T$, and once again, the value is observed as $\mathcal{L}^* = 3/5$ (this must coincide with the previous one because the randomized upper and lower values are the same!). ■

This procedure appears quite simple if there are only two actions per player. If $n = m = 100$, then the upper and lower envelopes are piecewise-linear functions in $\mathbb{R}^{99}$. This may be computationally impractical because all existing linear programming algorithms have running time at least exponential in dimension [264].

9.4 Nonzero-Sum Games

This section parallels the development of Section 9.3, except that the more general case of nonzero-sum games is considered. This enables games with any desired degree of conflict to be modeled. Some decisions may even benefit all players. One of the main applications of the subject is in economics, where it helps to explain the behavior of businesses in competition.

The saddle-point solution will be replaced by the *Nash equilibrium*, which again is based on eliminating regret. Since the players do not necessarily oppose each other, it is possible to model a game that involves any number of players. For nonzero games, new difficulties arise, such as the nonuniqueness of Nash equilibria and the computation of randomized Nash equilibria does not generally fit into

linear programming.

9.4.1 Two-Player Games

To help make the connection to Section 9.3 smoother, two-player games will be considered first. This case is also easier to understand because the notation is simpler. The ideas are then extended without difficulty from two players to many players. The game is formulated as follows.

Formulation 9.8 (A Two-Player Nonzero-Sum Game)

1. The same components as in Formulation 9.7, except the cost function.
2. A function, $L_1 : U \times V \rightarrow \mathbb{R} \cup \{\infty\}$, called the *cost function for* P_1.
3. A function, $L_2 : U \times V \rightarrow \mathbb{R} \cup \{\infty\}$, called the *cost function for* P_2.

The only difference with respect to Formulation 9.7 is that now there are two, independent cost functions, L_1 and L_2, one for each player. Each player would like to minimize its cost. There is no maximization in this problem; that appeared in zero-sum games because P_2 had opposing interests from P_1. A zero-sum game can be modeled under Formulation 9.7 by setting $L_1 = L$ and $L_2 = -L$.

Paralleling Section 9.3, first consider applying deterministic strategies to solve the game. As before, one possibility is that a player can apply its security strategy. To accomplish this, it does not even need to look at the cost function of the other player. It seems somewhat inappropriate, however, to neglect the consideration of both cost functions when making a decision. In most cases, the security strategy results in regret, which makes it inappropriate for nonzero-sum games.

A strategy that avoids regret will now be given. A pair (u^*, v^*) of actions is defined to be a *Nash equilibrium* if

$$L_1(u^*, v^*) = \min_{u \in U} \Big\{ L_1(u, v^*) \Big\} \tag{9.66}$$

and

$$L_2(u^*, v^*) = \min_{v \in V} \Big\{ L_2(u^*, v) \Big\}. \tag{9.67}$$

These expressions imply that neither P_1 nor P_2 has regret. Equation (9.66) indicates that P_1 is satisfied with its action, u^*, given the action, v^*, chosen by P_2. P_1 cannot reduce its cost any further by changing its action. Likewise, (9.67) indicates that P_2 is satisfied with its action v^*.

The game in Formulation 9.8 can be completely represented using two cost matrices. Let A and B denote the cost matrices for P_1 and P_2, respectively. Recall that Figure 9.2 showed a pattern for detecting a saddle point. A Nash equilibrium can be detected as shown in Figure 9.5. Think about the relationship between the two. If $A = -B$, then B can be negated and superimposed on top of A. This will yield the pattern in Figure 9.2 (each $\geq$ becomes $\leq$ because of

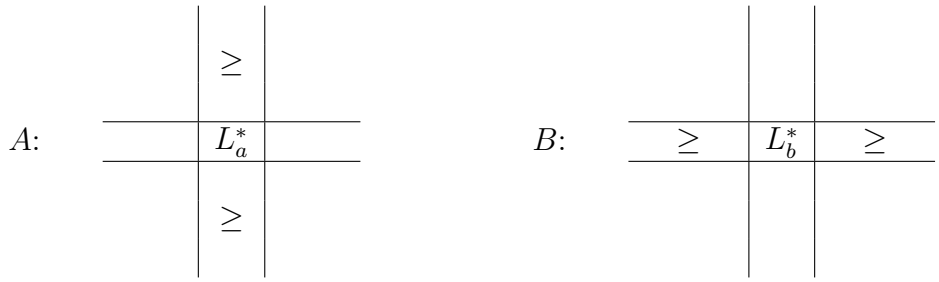

Figure 9.5: A Nash equilibrium can be detected in a pair of matrices by finding some (i, j) such that $L_a^* = L_1(i, j)$ is the lowest among all elements in column j of A, and $L_b^* = L_2(i, j)$ is the lowest among all elements in row i of B. Compare this with Figure 9.2.

negation). The values L_a^* and L_b^* coincide in this case. This observation implies that if $A = -B$, then the Nash equilibrium is actually the same concept as a saddle point. It applies, however, to much more general games.

Example 9.17 (A Deterministic Nash Equlibrium) Consider the game specified by the cost matrices A and B:

$$A: \quad \begin{array}{c|c|c|c} & V & & \\ U & 9 & 4 & 7 \\ & 6 & -1 & 5 \\ & 1 & 4 & 2 \end{array} \qquad B: \quad \begin{array}{c|c|c|c} & V & & \\ U & 2 & 1 & 6 \\ & 5 & 0 & 2 \\ & 2 & 2 & 5 \end{array} . \tag{9.68}$$

By applying (9.66) and (9.67), or by using the patterns in Figure 9.5, it can be seen that $u = 3$ and $v = 1$ is a Nash equilibrium. The resulting costs are $L_1 = 1$ and $L_2 = 2$. Another Nash equilibrium appears at $u = 2$ and $v = 2$. This yields costs $L_1 = -1$ and $L_2 = 0$, which is better for both players.

For zero-sum games, the existence of multiple saddle points did not cause any problem; however, for nonzero-sum games, there are great troubles. In the example shown here, one Nash equilibrium is clearly better than the other for both players. Therefore, it may seem reasonable that a rational DM would choose the better one. The issue of multiple Nash equilibria will be discussed next. ■

9.4.1.1 Dealing with multiple Nash equilibria

Example 9.17 was somewhat disheartening due to the existence of multiple Nash equilibria. In general, there could be any number of equilibria. How can each player know which one to play? If they each choose a different one, they are not guaranteed to fall into another equilibrium as in the case of saddle points of zero-sum games. Many of the equilibria can be eliminated by using Pareto optimality, which was explained in Section 9.1.1 and also appeared in Section 7.7.2 as a way

to optimally coordinate multiple robots. The idea is to formulate the selection as a multi-objective optimization problem, which fits into Formulation 9.2.

Consider two-dimensional vectors of the form (x_i, y_i), in which x and y represent the costs L_1 and L_2 obtained under the implementation of a Nash equilibrium denoted by π_i. For two different equilibria π_1 and π_2, the cost vectors (x_1, y_1) and (x_2, y_2) are obtained. In Example 9.17, these were $(1, 2)$ and $(-1, 0)$. In general, π_1 is said to be *better* than π_2 if $x_1 \leq x_2$, $y_1 \leq y_2$, and at least one of the inequalities is strict. In Example 9.17, the equilibrium that produces $(-1, 0)$ is clearly better than obtaining $(1, 2)$ because both players benefit.

The definition of "better" induces a partial ordering on the space of Nash equilibria. It is only partial because some vectors are incomparable. Consider, for example, $(-1, 1)$ and $(1, -1)$. The first one is preferable to P_1, and the second is preferred by P_2. In game theory, the Nash equilibria that are minimal with respect to this partial ordering are called *admissible*. They could alternatively be called *Pareto optimal*.

The best situation is when a game has one Nash equilibrium. If there are multiple Nash equilibria, then there is some hope that only one of them is admissible. In this case, it is hoped that the rational players are intelligent enough to figure out that any nonadmissible equilibria should be discarded. Unfortunately, there are many games that have multiple admissible Nash equilibria. In this case, analysis of the game indicates that the players must communicate or collaborate in some way to eliminate the possibility of regret. Otherwise, regret is unavoidable in the worst case. It is also possible that there are no Nash equilibria, but, fortunately, by allowing randomized strategies, a randomized Nash equilibrium is always guaranteed to exist. This will be covered after the following two examples.

Example 9.18 (The Battle of the Sexes) Consider a game specified by the cost matrices A and B:

$$A: \quad \begin{array}{c|cc} & V \\ U & \text{-2} & \text{0} \\ & \text{0} & \text{-1} \end{array} \qquad\qquad B: \quad \begin{array}{c|cc} & V \\ U & \text{-1} & \text{0} \\ & \text{0} & \text{-2} \end{array} . \tag{9.69}$$

This is a famous game called the "Battle of the Sexes." Suppose that a man and a woman have a relationship, and they each have different preferences on how to spend the evening. The man prefers to go shopping, and the woman prefers to watch a football match. The game involves selecting one of these two activities. The best case for either one is to do what they prefer while still remaining together. The worst case is to select different activities, which separates the couple. This game is somewhat unrealistic because in most situations some cooperation between them is expected.

Both $u = v = 1$ and $u = v = 2$ are Nash equilibria, which yield cost vectors $(-2, -1)$ and $(-1, -2)$, respectively. Neither solution is better than the other; therefore, they are both admissible. There is no way to avoid the possibility of regret unless the players cooperate (you probably already knew this). ■

The following is one of the most famous nonzero-sum games.

Example 9.19 (The Prisoner's Dilemma) The following game is very simple to express, yet it illustrates many interesting issues. Imagine that a heinous crime has been committed by two people. The authorities know they are guilty, but they do not have enough evidence to convict them. Therefore, they develop a plan to try to trick the suspects. Each suspect (or player) is placed in an isolated prison cell and given two choices. Each player can cooperate with the authorities, $u = 1$ or $v = 1$, or refuse, $u = 2$ or $v = 2$. By cooperating, the player admits guilt and turns over evidence to the authorities. By refusing, the player claims innocence and refuses to help the authorities.

The cost L_i represents the number of years that the player will be sentenced to prison. The cost matrices are assigned as

$$A: \quad \begin{array}{c|cc} U \backslash V & 1 & 2 \\ \hline 1 & 8 & 0 \\ 2 & 30 & 2 \end{array} \qquad B: \quad \begin{array}{c|cc} U \backslash V & 1 & 2 \\ \hline 1 & 8 & 30 \\ 2 & 0 & 2 \end{array} . \tag{9.70}$$

The motivation is that both players receive 8 years if they both cooperate, which is the sentence for being convicted of the crime and being rewarded for cooperating with the authorities. If they both refuse, then they receive 2 years because the authorities have insufficient evidence for a longer term. The interesting cases occur if one refuses and the other cooperates. The one who refuses is in big trouble because the evidence provided by the other will be used against him. The one who cooperates gets to go free (the cost is 0); however, the other is convicted on the evidence and spends 30 years in prison.

What should the players do? What would you do? If they could make a coordinated decision, then it seems that a good choice would be for both to refuse, which results in costs $(2, 2)$. In this case, however, there would be regret because each player would think that he had a chance to go free (receiving cost 0 by refusing). If they were to play the game a second time, they might be inclined to change their decisions.

The Nash equilibrium for this problem is for both of them to cooperate, which results in $(8, 8)$. Thus, they pay a price for not being able to communicate and coordinate their strategy. This solution is also a security strategy for the players, because it achieves the lowest cost using worst-case analysis. ■

9.4.1.2 Randomized Nash equilibria

What happens if a game has no Nash equilibrium over the space of deterministic strategies? Once again the problem can be alleviated by expanding the strategy space to include randomized strategies. In Section 9.3.3 it was explained that every zero-sum game under Formulation 9.7 has a randomized saddle point on the space of randomized strategies. It was shown by Nash that every nonzero-sum

game under Formulation 9.8 has a randomized Nash equilibrium [730]. This is a nice result; however, there are a couple of concerns. There may still exist other admissible equilibria, which means that there is no reliable way to avoid regret unless the players collaborate. The other concern is that randomized Nash equilibria unfortunately cannot be computed using the linear programming approach of Section 9.3.3. The required optimization is instead a form of nonlinear programming [94, 664, 731], which does not necessarily admit a nice, combinatorial solution.

Recall the definition of randomized strategies from Section 9.3.3. For a pair (w, z) of randomized strategies, the expected costs, $\bar{L}^1(w, z)$ and $\bar{L}^2(w, z)$, can be computed using (9.57). A pair (w^*, z^*) of strategies is said to be a *randomized Nash equilibrium* if

$$\bar{L}^1(w^*, z^*) = \min_{w \in W} \left\{ \bar{L}^1(w, z^*) \right\} \tag{9.71}$$

and

$$\bar{L}^2(w^*, z^*) = \min_{z \in Z} \left\{ \bar{L}^2(w^*, z) \right\}. \tag{9.72}$$

In game-theory literature, this is usually referred to as a *mixed Nash equilibrium*. Note that (9.71) and (9.72) are just generalizations of (9.66) and (9.67) from the space of deterministic strategies to the space of randomized strategies.

Using the cost matrices A and B, the Nash equilibrium conditions can be written as

$$w^* A z^* = \min_{w \in W} \left\{ w A z^* \right\} \tag{9.73}$$

and

$$w^* B z^* = \min_{z \in Z} \left\{ w^* B z \right\}. \tag{9.74}$$

Unfortunately, the computation of randomized Nash equilibria is considerably more challenging than computing saddle points. The main difficulty is that Nash equilibria are not necessarily security strategies. By using security strategies, it is possible to decouple the decisions of the players into separate linear programming problems, as was seen in Example 9.16. For the randomized Nash equilibrium, the optimization between the players remains coupled. The resulting optimization is often referred to as the *linear complementarity problem*. This can be formulated as a nonlinear programming problem [664, 731], which means that it is a nonlinear optimization that involves both equality and inequality constraints on the domain (in this particular case, a *bilinear programming* problem is obtained [59]).

Example 9.20 (Finding a Randomized Nash Equilibrium) To get an idea of the kind of optimization that is required, recall Example 9.18. A randomized Nash equilibrium that is distinct from the two deterministic equilibria can be found. Using the cost matrices from Example 9.18, the expected cost for P_1 given

randomized strategies w and z is

$$\begin{aligned} \bar{L}^1(w,z) &= wAz \\ &= \begin{pmatrix} w_1 & w_2 \end{pmatrix} \begin{pmatrix} -2 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \\ &= -2w_1z_1 - w_2z_2 \\ &= -3w_1z_1 + w_1 + z_1, \end{aligned} \tag{9.75}$$

in which the final step uses the fact that $w_2 = 1 - w_1$ and $z_2 = 1 - z_1$. Similarly, the expected cost for P_2 is

$$\begin{aligned} \bar{L}^2(w,z) &= wBz \\ &= \begin{pmatrix} w_1 & w_2 \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & -2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \\ &= -w_1z_1 - 2w_2z_2 \\ &= -3w_1z_1 + 2w_1 + 2z_1. \end{aligned} \tag{9.76}$$

If z is fixed, then the final equation in (9.75) is linear in w; likewise, if w is fixed, then (9.76) is linear in z. In the case of computing saddle points for zero-sum games, we were allowed to make this assumption; however, it is not possible here. We must choose (w^*, z^*) to simultaneously optimize (9.75) while $z = z^*$ and (9.76) while $w = w^*$.

It turns out that this problem is simple enough to solve with calculus. Using the classical optimization method of taking derivatives, a candidate solution can be found by computing

$$\frac{\partial \bar{L}^1(w_1, z_1)}{\partial w_1} = 1 - 3z_1 \tag{9.77}$$

and

$$\frac{\partial \bar{L}^2(w_1, z_1)}{\partial z_1} = 2 - 3w_1. \tag{9.78}$$

Extrema occur when both of these simultaneously become 0. Solving $1 - 3z_1 = 0$ and $2 - 3w_1 = 0$ yields $(w^*, z^*) = (2/3, 1/3)$, which is a randomized Nash equilibrium. The deterministic Nash equilibria are not detected by this method because they occur on the boundary of W and Z, where the derivative is not defined. ■

The computation method in Example 9.20 did not appear too difficult because there were only two actions per player, and half of the matrix costs were 0. In general, two complicated equations must be solved simultaneously. The expressions, however, are always second-degree polynomials. Furthermore, they each become linear with respect to the other variables if w or z is held fixed.

Summary of possible solutions The solution possibilities to remember for a nonzero-sum game under Formulation 9.8 are as follows.

1. There may be multiple, admissible (deterministic) Nash equilibria.
2. There may be no (deterministic) Nash equilibria.
3. There is always at least one randomized Nash equilibrium.

9.4.2 More Than Two Players

The ideas of Section 9.4.1 easily generalize to any number of players. The main difficulty is that complicated notation makes the concepts appear more difficult. Keep in mind, however, that there are no fundamental differences. A nonzero-sum game with n players is formulated as follows.

Formulation 9.9 (An n-Player Nonzero-Sum Game)

1. A set of n players, P_1, P_2, ..., P_n.
2. For each player P_i, a finite, nonempty set U^i called the *action space* for P_i. For convenience, assume that each U^i is a set of consecutive integers from 1 to $|U^i|$. Each $u^i \in U^i$ is referred to as an *action* of P_i.
3. For each player P_i, a function, $L_i : U^1 \times U^2 \times \cdots \times U^n \rightarrow \mathbb{R} \cup \{\infty\}$ called the *cost function for* P_i.

A matrix formulation of the costs is no longer possible because there are too many dimensions. For example, if $n = 3$ and $|U^i| = 2$ for each player, then $L_i(u^1, u^2, u^3)$ is specified by a $2 \times 2 \times 2$ cube of 8 entries. Three of these cubes are needed to specify the game. Thus, it may be helpful to just think of L_i as a multivariate function and avoid using matrices.[4]

The Nash equilibrium idea generalizes by requiring that each P_i experiences no regret, given the actions chosen by the other $n - 1$ players. Formally, a set $(u^{1*}, \ldots, u^{n*})$ of actions is said to be a (deterministic) *Nash equilibrium* if

$$L_i(u^{1*}, \ldots, u^{i*}, \ldots, u^{n*}) = \min_{u^i \in U^i} \left\{ L_i(u^{1*}, \ldots, u^{(i-1)*}, u^i, u^{(i+1)*}, \ldots, u^{n*}) \right\} \quad (9.79)$$

for every $i \in \{1, \ldots, n\}$.

For $n > 2$, any of the situations summarized at the end of Section 9.4.1 can occur. There may be no deterministic Nash equilibria or multiple Nash equilibria. The definition of an admissible Nash equilibrium is extended by defining the notion of *better* over n-dimensional cost vectors. Once again, the minimal vectors

[4] If you enjoy working with tensors, these could be used to capture n-player cost functions [107].

with respect to the resulting partial ordering are considered *admissible* (or *Pareto optimal*). Unfortunately, multiple admissible Nash equilibria may still exist.

It turns out that for any game under Formulation 9.9, there exists a randomized Nash equilibrium. Let z^i denote a randomized strategy for P_i. The expected cost for each P_i can be expressed as

$$\bar{L}^i(z^1, z^2, \ldots, z^n) = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \cdots \sum_{i_n=1}^{m_n} L_i(i_1, i_2, \ldots, i_n) z^1_{i_1} z^2_{i_2} \cdots z^n_{i_n}. \tag{9.80}$$

Let Z^i denote the space of randomized strategies for P_i. An assignment, $(z^{1*}, \ldots, z^{n*})$, of randomized strategies to all of the players is called a *randomized Nash equilibrium* if

$$\bar{L}^i(z^{1*}, \ldots, z^{i*}, \ldots, z^{n*}) = \min_{z^i \in Z^i} \left\{ \bar{L}^i(z^{1*}, \ldots, z^{(i-1)*}, z^i, z^{(i+1)*}, \ldots, z^{n*}) \right\} \tag{9.81}$$

for all $i \in \{1, \ldots, n\}$.

As might be expected, computing a randomized Nash equilibrium for $n > 2$ is even more challenging than for $n = 2$. The method of Example 9.20 can be generalized to n-player games; however, the expressions become even more complicated. There are n equations, each of which appears linear if the randomized strategies are fixed for the other $n-1$ players. The result is a collection of n-degree polynomials over which n optimization problems must be solved simultaneously.

Example 9.21 (A Three-Player Nonzero-Sum Game) Suppose there are three players, P_1, P_2, and P_3, each of which has two actions, 1 and 2. A deterministic strategy is specified by a vector such as $(1, 2, 1)$, which indicates $u^1 = 1$, $u^2 = 2$, and $u^3 = 1$.

Now some costs will be defined. For convenience, let

$$L(i, j, k) = \Big(L_1(i, j, k),\ L_2(i, j, k),\ L_3(i, j, k) \Big) \tag{9.82}$$

for each $i, j, k \in \{1, 2\}$. Let the costs be

$$\begin{aligned}
L(1,1,1) &= (1, 1, -2) & L(1,1,2) &= (-4, 3, 1) \\
L(1,2,1) &= (2, -4, 2) & L(1,2,2) &= (-5, -5, 10) \\
L(2,1,1) &= (3, -2, -1) & L(2,1,2) &= (-6, -6, 12) \\
L(2,2,1) &= (2, 2, -4) & L(2,2,2) &= (-2, 3, -1).
\end{aligned} \tag{9.83}$$

There are two deterministic Nash equilibria, which yield the costs $(2, -4, 2)$ and $(3, -2, -1)$. These can be verified using (9.79). Each player is satisfied with the outcome given the actions chosen by the other players. Unfortunately, both Nash equilibria are both admissible. Therefore, some collaboration would be needed between the players to ensure that no regret will occur. ■

9.5 Decision Theory Under Scrutiny

Numerous models for decision making were introduced in this chapter. These provide a foundation for planning under uncertainty, which is the main focus of Part III. Before constructing planning models with this foundation, it is important to critically assess how appropriate it may be in applications. You may have had many questions while reading Sections 9.1 to 9.4. How are the costs determined? Why should we believe that optimizing the *expected* cost is the right thing to do? What happens if prior probability distributions are not available? Is worst-case analysis too conservative? Can we be sure that players in a game will follow the assumed rational behavior? Is it realistic that players know each other's cost functions? The purpose of this section is to help shed some light on these questions. A building is only as good as its foundation. Any mistakes made by misunderstanding the limitations of decision theory will ultimately work their way into planning formulations that are constructed from them.

9.5.1 Utility Theory and Rationality

This section provides some justification for using cost functions and then minimizing their expected value under Formulations 9.3 and 9.4. The resulting framework is called *utility theory*, which is usually formulated using rewards instead of costs. As stated in Section 9.1.1, a cost can be converted into a reward by multiplying by -1 and then swapping each maximization with minimization. We will therefore talk about a reward R with the intuition that a higher reward is better.

9.5.1.1 Comparing rewards

Imagine assigning reward values to various outcomes of a decision-making process. In some applications numerical values may come naturally. For example, the reward might be the amount of money earned in a financial investment. In robotics applications, one could negate time to execute a task or the amount of energy consumed. For example, the reward could indicate the amount of remaining battery life after a mobile robot builds a map.

In some applications the source of rewards may be subjective. For example, what is the reward for washing dishes, in comparison to sweeping the floor? Each person would probably assign different rewards, which may even vary from day to day. It may be based on their enjoyment or misery in performing the task, the amount of time each task would take, the perceptions of others, and so on. If decision theory is used to automate the decision process for a human "client," then it is best to consult carefully with the client to make sure you know their preferences. In this situation, it may be possible to sort their preferences and then assign rewards that are consistent with the ordering.

Once the rewards are assigned, consider making a decision under Formulation 9.1, which does not involve nature. Each outcome corresponds directly to an action, $u \in U$. If the rewards are given by $R : U \rightarrow \mathbb{R}$, then the cost, L, can be

defined as $L(u) = -R(u)$ for every $u \in U$. Satisfying the client is then a matter of choosing u to minimize L.

Now consider a game against nature. The decision now involves comparing probability distributions over the outcomes. The space of all probability distributions may be enormous, but this is simplified by using expectation to map each probability distribution (or density) to a real value. The concern should be whether this projection of distributions onto real numbers will fail to reflect the true preferences of the client. The following example illustrates the effect of this.

Example 9.22 (Do You Like to Gamble?) Suppose you are given three choices:

1. You can have 1000 Euros.

2. We will toss an unbiased coin, and if the result is heads, then you will receive 2000 Euros. Otherwise, you receive nothing.

3. With probability 2/3, you can have 3000 Euros; however, with probability 1/3, you have to give me 3000 Euros.

The expected reward for each of these choices is 1000 Euros, but would you really consider these to be equivalent? Your love or disdain for gambling is not being taken into account by the expectation. How should such an issue be considered in games against nature? ■

To begin to fix this problem, it is helpful to consider another scenario. Many people would probably agree that having more money is preferable (if having too much worries you, then you can always give away the surplus to your favorite charities). What is interesting, however, is that being wealthy decreases the perceived value of money. This is illustrated in the next example.

Example 9.23 (Reality Television) Suppose you are lucky enough to appear on a popular reality television program. The point of the show is to test how far you will go in making a fool out of yourself, or perhaps even torturing yourself, to earn some money. You are asked to do some unpleasant task (such as eating cockroaches, or holding your head under water for a long time, and so on.). Let u_1 be the action to agree to do the task, and let u_2 mean that you decline the opportunity. The prizes are expressed in U.S. dollars. Imagine that you are a starving student on a tight budget.

Below are several possible scenarios that could be presented on the television program. Consider how you would react to each one.

1. Suppose that u_1 earns you \$1 and u_2 earns you nothing. Purely optimizing the reward would lead to choosing u_1, which means performing the unpleasant task. However, is this worth \$1? The problem so far is that we are not taking into account the amount of discomfort in completing a task. Perhaps it might make sense to make a reward function that shifts the dollar values

by subtracting the amount for which you would be just barely willing to perform the task.

2. Suppose that u_1 earns you \$10,000 and u_2 earns you nothing. \$10,000 is assumed to be an enormous amount of money, clearly worth enduring any torture inflicted by the television program. Thus, u_1 is preferable.

3. Now imagine that the television host first gives you \$10 million just for appearing on the program. Are you still willing to perform the unpleasant task for an extra \$10,000? Probably not. What is happening here? Your sense of value assigned to money seems to decrease as you get more of it, right? It would not be too interesting to watch the program if the contestants were all wealthy oil executives.

4. Suppose that you have performed the task and are about to win the prize. Just to add to the drama, the host offers you a gambling opportunity. You can select action u_1 and receive \$10,000, or be a gambler by selecting u_2 and have probability 1/2 of winning \$25,000 by the tossing of a fair coin. In terms of the expected reward, the clear choice is u_2. However, you just completed the unpleasant task and expect to earn money. The risk of losing it all may be intolerable. Different people will have different preferences in this situation.

5. Now suppose once again that you performed the task. This time your choices are u_1, to receive \$100, or u_2, to have probability 1/2 of receiving \$250 by tossing a fair coin. The host is kind enough, though, to let you play 100 times. In this case, the expected totals for the two actions are \$10,000 and \$12,500, respectively. This time it seems clear that the best choice is to gamble. After 100 independent trials, we would expect that, with extremely high probability, over \$10,000 would be earned. Thus, reasoning by expected-case analysis seems valid if we are allowed numerous, independent trials. In this case, with high probability a value close to the expected reward should be received.

■

Based on these examples, it seems that the client or evaluator of the decision-making system must indicate preferences between probability distributions over outcomes. There is a formal way to ensure that once these preferences are assigned, a cost function can be designed for which its expectation faithfully reflects the preferences over distributions. This results in *utility theory*, which involves the following steps:

1. Require that the client is *rational* when assigning preferences. This notion is defined through axioms.

2. If the preferences are assigned in a way that is consistent with the axioms, then a utility function is guaranteed to exist. When expected utility is optimized, the preferences match exactly those of the client.

3. The cost function can be derived from the utility function.

The client must specify preferences among probability distributions of outcomes. Suppose that Formulation 9.2 is used. For convenience, assume that U and Θ are finite. Let X denote a *state space* based on outcomes.[5] Let $f : U \times \Theta \rightarrow X$ denote a mapping that assigns a state to every outcome. A simple example is to declare that $X = U \times \Theta$ and make f the identity map. This makes the outcome space and state space coincide. It may be convenient, though, to use f to collapse the space of outcomes down to a smaller set. If two outcomes map to the same state using f, then it means that the outcomes are indistinguishable as far as rewards or costs are concerned.

Let z denote a probability distribution over X, and let Z denote the set of all probability distributions over X. Every $z \in Z$ is represented as an n-dimensional vector of probabilities in which $n = |X|$; hence, it is considered as an element of $\mathbb{R}^n$. This makes it convenient to "blend" two probability distributions. For example, let $\alpha \in (0, 1)$ be a constant, and let z_1 and z_2 be any two probability distributions. Using scalar multiplication, a new probability distribution, $\alpha z_1 + (1 - \alpha)z_2$, is obtained, which is a *blend* of z_1 and z_2. Conveniently, there is no need to normalize the result. It is assumed that z_1 and z_2 initially have unit magnitude. The blend has magnitude $\alpha + (1 - \alpha) = 1$.

The modeler of the decision process must consult the client to represent preferences among elements of Z. Let $z_1 \prec z_2$ mean that z_2 is strictly preferred over z_1. Let $z_1 \approx z_2$ mean that z_1 and z_2 are equivalent in preference. Let $z_1 \preceq z_2$ mean that either $z_1 \prec z_2$ or $z_1 \approx z_2$. The following example illustrates the assignment of preferences.

Example 9.24 (Indicating Preferences) Suppose that $U = \Theta = \{1, 2\}$, which leads to four possible outcomes: $(1, 1)$, $(1, 2)$, $(2, 1)$, and $(2, 2)$. Imagine that nature represents a machine that generates 1 or 2 according to a probability distribution. The action is to guess the number that will be generated by the machine. If you pick the same number, then you win that number of gold pieces. If you do not pick the same number, then you win nothing, but also lose nothing.

Consider the construction of the state space X by using f. The outcomes $(2, 1)$ and $(1, 2)$ are identical concerning any conceivable reward. Therefore, these should map to the same state. The other two outcomes are distinct. The state space therefore needs only three elements and can be defined as $X = \{0, 1, 2\}$. Let $f(2, 1) = f(1, 2) = 0$, $f(1, 1) = 1$, and $f(2, 2) = 2$. Thus, the last two states indicate that some gold will be earned.

The set Z of probability distributions over X is now considered. Each $z \in Z$ is a three-dimensional vector. As an example, $z_1 = [1/2 \;\; 1/4 \;\; 1/4]$ indicates that the

[5] In most utility theory literature, this is referred to as a *reward space*, $\mathcal{R}$ [89].

state will be 0 with probability 1/2, 1 with probability 1/4, and 2 with probability 1/4. Suppose $z_2 = [1/3 \;\; 1/3 \;\; 1/3]$. Which distribution would you prefer? It seems in this case that z_2 is uniformly better than z_1 because there is a greater chance of winning gold. Thus, we declare $z_1 \prec z_2$. The distribution $z_3 = [1 \;\; 0 \;\; 0]$ seems to be the worst imaginable. Hence, we can safely declare $z_3 \prec z_1$ and $z_1 \prec z_2$.

The procedure of determining the preferences can become quite tedious for complicated problems. In the current example, Z is a 2D subset of $\mathbb{R}^3$. This subset can be partitioned into a finite set of regions over which the client may be able to clearly indicate preferences. One of the major criticisms of this framework is the impracticality of determining preferences over Z [831].

After the preferences are determined, is there a way to ensure that a real-value function on X exists for which the expected value exactly reflects the preferences? If the axioms of rationality are satisfied by the assignment of preferences, then the answer is *yes*. These axioms are covered next. ■

9.5.1.2 Axioms of rationality

To meet the goal of designing a utility function, it turns out that the preferences must follow rules called the *axioms of rationality*. They are sensible statements of consistency among the preferences. As long as these are followed, then a utility function is guaranteed to exist (detailed arguments appear in [268, 831]). The decision maker is considered *rational* if the following axioms are followed when defining $\prec$ and $\approx$:[6]

1. If $z_1, z_2 \in Z$, then either $z_1 \preceq z_2$ or $z_2 \preceq z_1$.
   "You must be able to make up your mind."

2. If $z_1 \preceq z_2$ and $z_2 \preceq z_3$, then $z_1 \preceq z_3$.
   "Preferences must be transitive."

3. If $z_1 \prec z_2$, then
   $$\alpha z_1 + (1 - \alpha) z_3 \prec \alpha z_2 + (1 - \alpha) z_3, \tag{9.84}$$
   for any $z_3 \in Z$ and $\alpha \in (0, 1)$.
   "Evenly blending in a new distribution does not alter preference."

4. If $z_1 \prec z_2 \prec z_3$, then there exists some $\alpha \in (0, 1)$ and $\beta \in (0, 1)$ such that
   $$\alpha z_1 + (1 - \alpha) z_3 \prec z_2 \tag{9.85}$$
   and
   $$z_2 \prec \beta z_1 + (1 - \beta) z_3. \tag{9.86}$$
   "There is no heaven or hell."

[6] Alternative axiom systems exist [268, 839].

Each axiom has an intuitive interpretation that makes practical sense. The first one simply indicates that the preference direction can always be inferred for a pair of distributions. The second axiom indicates that preferences must be transitive.[7] The last two axioms are somewhat more complicated. In the third axiom, z_2 is strictly preferred to z_1. An attempt is made to cause confusion by blending in a third distribution, z_3. If the same "amount" of z_3 is blended into both z_1 and z_2, then the preference should not be affected. The final axiom involves z_1, z_2, and z_3, each of which is strictly better than its predecessor. The first equation, (9.85), indicates that if z_2 is strictly better than z_1, then a tiny amount of z_3 can be blended into z_1, with z_2 remaining preferable. If z_3 had been like "heaven" (i.e., infinite reward), then this would not be possible. Similarly, (9.86) indicates that a tiny amount of z_1 can be blended into z_3, and the result remains better than z_2. This means that z_1 cannot be "hell," which would have infinite negative reward.[8]

9.5.1.3 Constructing a utility function

If the preferences have been determined in a way consistent with the axioms, then it can be shown that a *utility function* always exists. This means that there exists a function $\mathcal{U} : X \rightarrow \mathbb{R}$ such that, for all $z_1, z_2 \in Z$,

$$z_1 \prec z_2 \text{ if and only if } E_{z_1}[\mathcal{U}] < E_{z_2}[\mathcal{U}], \tag{9.87}$$

in which E_{z_i} denotes the expected value of $\mathcal{U}$, which is being treated as a random variable under the probability distribution z_i. The existence of $\mathcal{U}$ implies that it is safe to determine the best action by maximizing the expected utility.

A reward function can be defined using a utility function, $\mathcal{U}$, as $R(u, \theta) = \mathcal{U}(f(u, \theta))$. The utility function can be converted to a cost function as $L(u, \theta) = -R(u, \theta) = -\mathcal{U}(f(u, \theta))$. Minimizing the expected cost, as was recommended under Formulations 9.3 and 9.4 with probabilistic uncertainty, now seems justified under the assumption that $\mathcal{U}$ was constructed correctly to preserve preferences.

Unfortunately, establishing the existence of a utility function does not produce a systematic way to construct it. In most circumstances, one is forced to design $\mathcal{U}$ by a trial-and-error process that involves repeatedly checking the preferences. In the vast majority of applications, people create utility and cost functions without regard to the implications discussed in this section. Thus, undesirable conclusions may be reached in practice. Therefore, it is important not to be too confident about the quality of an optimal decision rule.

Note that if worst-case analysis had been used, then most of the problems discussed here could have been avoided. Worst-case analysis, however, has its weaknesses, which will be discussed in Section 9.5.3.

[7] For some reasonable problems, however, transitivity is not desirable. See the Candorcet and Simpson paradoxes in [831].

[8] Some axiom systems allow infinite rewards, which lead to utility and cost functions with infinite values, but this is not considered here.

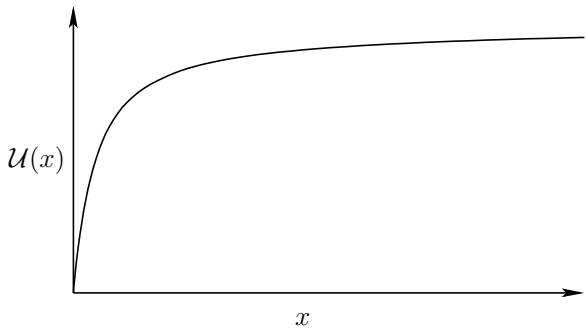

Figure 9.6: The utility of new amounts of money decreases as the total accumulation of wealth increases. The utility function may even bounded.

Example 9.25 (The Utility of Money) We conclude the section by depicting a utility function that is often applied to money. Suppose that the state space $X = \mathbb{R}$, which corresponds to the amount of U.S. dollars earned. The utility of money applied by most people indicates that the value of new increments of money decreases as the total accumulated wealth increases. The utility function may even be bounded. Imagine there is some maximum dollar amount, such as $\$10^{100}$, after which additional money has no value. A typical utility curve is shown in Figure 9.6 [89]. ■

9.5.2 Concerns Regarding the Probabilistic Model

Section 9.5.1 addressed the source of cost functions and the validity of taking their expectations. This section raises concerns over the validity of the probability distributions used in Section 9.2. The two main topics are criticisms of Bayesian methods in general and problems with constructing probability distributions.

9.5.2.1 Bayesians vs. frequentists

For the past century and a half, there has been a fundamental debate among statisticians on the *meaning* of probabilities. Virtually everyone is satisfied with the axioms of probability, but beyond this, what is their meaning when making inferences? The two main camps are the *frequentists* and the *Bayesians*. A form of Bayes' rule was published in 1763 after the death of Bayes [80]. During most of the nineteenth century Bayesian analysis tended to dominate literature; however, during the twentieth century, the frequentist philosophy became more popular as a more rigorous interpretation of probabilities. In recent years, the credibility of Bayesian methods has been on the rise again.

As seen so far, a Bayesian interprets probabilities as the degree of belief in a hypothesis. Under this philosophy, it is perfectly valid to begin with a prior

distribution, gather a few observations, and then make decisions based on the resulting posterior distribution from applying Bayes' rule.

From a frequentist perspective, Bayesian analysis makes far too liberal use of probabilities. The frequentist believes that probabilities are only defined as the quantities obtained in the limit after the number of independent trials tends to infinity. For example, if an unbiased coin is tossed over numerous trials, the probability 1/2 represents the value to which the ratio between heads and the total number of trials will converge as the number of trials tends to infinity. On the other hand, a Bayesian might say that the probability that the *next* trial results in heads is 1/2. To a frequentist, this interpretation of probability is too strong.

Frequentists have developed a version of decision theory based on their philosophy; comparisons between the two appear in [831]. As an example, a frequentist would advocate optimizing the following *frequentist risk* to obtain a decision rule:

$$R(\theta, \pi) = \int_y L(\pi(y), \theta) P(y|\theta) dy, \tag{9.88}$$

in which π represents the strategy, $\pi : Y \rightarrow U$. The frequentist risk averages over all data, rather than making a decision based on a single observation, as advocated by Bayesians in (9.26). The probability $P(y|\theta)$ is assumed to be obtained in the limit as the number of independent data trials tends to infinity. The main drawback in using (9.88) is that the optimization depends on θ. The resulting best decision rule must depend on θ, which is unknown. In some limited cases, it may be possible to select some π that optimizes (9.88) for all θ, but this rarely occurs. Thus, the frequentist risk can be viewed as a constraint on the desirability of strategies, but it usually is not powerful enough to select a single one. This problem is reminiscent of Pareto optimality, which was discussed in Section 9.1.1. The frequentist approach attempts to be more conservative and rigorous, with the result being that weaker statements are made regarding decisions.

9.5.2.2 The source of prior distributions

Suppose that the Bayesian method has been adopted. The most widespread concern in all Bayesian analyses is the source of the prior distribution. In Section 9.2, this is represented as $P(\theta)$ (or $p(\theta)$), which represents a distribution (or density) over the nature action space. The best way to obtain $P(\theta)$ is by estimating the distribution over numerous independent trials. This brings its definition into alignment with frequentist views. This was possible with Example 9.11, in which $P(\theta)$ could be reliably estimated from the frequency of occurrence of letters across numerous pages of text. The distribution could even be adapted to a particular language or theme.

In most applications that use decision theory, however, it is impossible or too costly to perform such experiments. What should be done in this case? If a prior distribution is simply "made up," then the resulting posterior probabilities may be suspect. In fact, it may be invalid to call them probabilities at all. Sometimes

the term *subjective probabilities* is used in this case. Nevertheless, this is commonly done because there are few other options. One of these options is to resort to frequentist decision theory, but, as mentioned, it does not work with single observations.

Fortunately, as the number of observations increases, the influence of the prior on the Bayesian posterior distributions diminishes. If there is only one observation, or even none as in Formulation 9.3, then the prior becomes very influential. If there is little or no information regarding $P(\theta)$, the distribution should be designed as carefully as possible. It should also be understood that whatever conclusions are made with this assumption, they are biased by the prior. Suppose this model is used as the basis of a planning approach. You might feel satisfied computing the "optimal" plan, but this notion of optimality could still depend on some arbitrary initial bias due to the assignment of prior values.

If there is no information available, then it seems reasonable that $P(\theta)$ should be as uniform as possible over Θ. This was referred to by Laplace as the "principle of insufficient reason" [581]. If there is no reason to believe that one element is more likely than another, then they should be assigned equal values. This can also be justified by using Shannon's entropy measure from information theory [49, 248, 864]. In the discrete case, this is

$$-\sum_{\theta \in \Theta} P(\theta) \lg P(\theta), \tag{9.89}$$

and in the continuous case it is

$$-\int_{\Theta} p(\theta) \lg p(\theta) d\theta. \tag{9.90}$$

This entropy measure was developed in the context of communication systems to estimate the minimum number of bits needed to encode messages delivered through a noisy medium. It generally indicates the amount of uncertainty associated with the distribution. A larger value of entropy implies a greater amount of uncertainty.

It turns out that the entropy function is maximized when $P(\theta)$ is a uniform distribution, which seems to justify the principle of insufficient reason. This can be considered as a *noninformative prior*. The idea is even applied quite frequently when $\Theta = \mathbb{R}$, which leads to an *improper prior*. The density function cannot maintain a constant, nonzero value over all of $\mathbb{R}$ because its integral would be infinite. Since the decisions made in Section 9.2 do not depend on any normalizing factors, a constant value can be assigned for $p(\theta)$ and the decisions are not affected by the fact that the prior is improper.

The main difficulty with applying the entropy argument in the selection of a prior is that Θ itself may be chosen in a number of arbitrary ways. Uniform assignments to different choices of Θ ultimately yield different information regarding the priors. Consider the following example.

Example 9.26 (A Problem with Noninformative Priors) Consider a decision about what activities to do based on the weather. Imagine that there is absolutely no information about what kind of weather is possible. One possible assignment is $\Theta = \{p, c\}$, in which p means "precipitation" and c means "clear." Maximizing (9.89) suggests assigning $P(p) = P(c) = 1/2$.

After thinking more carefully, perhaps we would like to distinguish between different kinds of precipitation. A better set of nature actions would be $\Theta = \{r, s, c\}$, in which c still means "clear," but precipitation p has been divided into r for "rain" and s for "snow." Now maximizing (9.89) assigns probability 1/3 to each nature action. This is clearly different from the original assignment. Now that we distinguish between different kinds of precipitation, it seems that precipitation is much more likely to occur. Does our preference to distinguish between different forms of precipitation really affect the weather? ■

Example 9.27 (Noninformitive Priors for Continuous Spaces) Similar troubles can result in continuous spaces. Recall the parameter estimation problem described in Example 9.12. Suppose instead that the task is to estimate a line based on some data points that were supposed to fall on the line but missed due to noise in the measurement process.

What initial probability density should be assigned to Θ, the set of all lines? Suppose that the line lives in $Z = \mathbb{R}^2$. The line equation can be expressed as

$$\theta_1 z_1 + \theta_2 z_2 + \theta_3 = 0. \tag{9.91}$$

The problem is that if the parameter vector, $\theta = [\theta_1 \quad \theta_2 \quad \theta_3]$, is multiplied by a scalar constant, then the same line is obtained. Thus, even though $\theta \in \mathbb{R}^3$, a constraint must be added. Suppose we require that

$$\theta_1^2 + \theta_2^2 + \theta_3^1 = 1 \tag{9.92}$$

and $\theta_1 \geq 0$. This mostly fixes the problem and ensures that each parameter value corresponds to a unique line (except for some duplicate cases at $\theta_1 = 0$, but these can be safely neglected here). Thus, the parameter space is the upper half of a sphere, $\mathbb{S}^2$. The maximum-entropy prior suggests assigning a uniform probability density to Θ. This may feel like the right thing to do, but this notion of uniformity is biased by the particular constraint applied to the parameter space to ensure uniqueness. There are many other choices. For example, we could replace (9.92) by constraints that force the points to lie on the upper half of the surface of cube, instead of a sphere. A uniform probability density assigned in this new parameter space certainly differs from one over the sphere.

In some settings, there is a natural representation of the parameter space that is invariant to certain transformations. Section 5.1.4 introduced the notion of Haar measure. If the Haar measure is used as a noninformative prior, then a meaningful notion of uniformity may be obtained. For example, suppose that the parameter space is $SO(3)$. Uniform probability mass over the space of unit quaternions,

as suggested in Example 5.14, is an excellent choice for a noninformative prior because it is consistent with the Haar measure, which is invariant to group operations applied to the events. Unfortunately, a Haar measure does not exist for most spaces that arise in practice.[9] ■

9.5.2.3 Incorrect assumptions on conditional distributions

One final concern is that many times even the distribution $P(y|\theta)$ is incorrectly estimated because it is assumed arbitrarily to belong to a family of distributions. For example, it is often very easy to work with Gaussian densities. Therefore, it is tempting to assume that $p(y|\theta)$ is Gaussian. Experiments can be performed to estimate the mean and variance parameters. Even though some best fit will be found, it does not necessarily imply that a Gaussian is a good representation. Conclusions based on this model may be incorrect, especially if the true distribution has a different shape, such as having a larger tail or being multimodal. In many cases, *nonparametric* methods may be needed to avoid such biases. Such methods do not assume a particular family of distributions. For example, imagine estimating a probability distribution by making a histogram that records the frequency of y occurrences for a fixed value of θ. The histogram can then be normalized to contain a representation of the probability distribution without assuming an initial form.

9.5.3 Concerns Regarding the Nondeterministic Model

Given all of the problems with probabilistic modeling, it is tempting to abandon the whole framework and work strictly with the nondeterministic model. This only requires specifying Θ, without indicating anything about the relative likelihoods of various actions. Therefore, most of the complicated issues presented in Sections 9.5.1 and 9.5.2 vanish. Unfortunately, this advantage comes at a substantial price. Making decisions with worst-case analysis under the nondeterministic model has its own shortcomings. After considering the trade-offs, you can decide which is most appropriate for a particular application of interest.

The first difficulty is to ensure that Θ is sufficiently large to cover all possibilities. Consider Formulation 9.6, in which nature acts twice. Through a nature observation action space, $\Psi(\theta)$, interference is caused in the measurement process. Suppose that $\Theta = \mathbb{R}$ and $h(\theta, \psi) = \theta + \psi$. In this case, $\Psi(\theta)$ can be interpreted as the measurement error. What is the maximum amount of error that can occur? Perhaps a sonar is measuring the distance from the robot to a wall. Based on the sensor specifications, it may be possible to construct a nice bound on the error. Occasionally, however, the error may be larger than this bound. Sonars sometimes fail to hear the required echo to compute the distance. In this case the

[9] A locally compact topological group is required [346, 836].

reported distance is ∞. Due to reflections, extremely large errors can sometimes occur. Although such errors may be infrequent, if we want *guaranteed* performance, then large or even infinite errors should be included in $\Psi(\theta)$. The problem is that worst-case reasoning could always conclude that the sensor is useless by reporting ∞. Any statistically valid information that could be gained from the sensor would be ignored. Under the probabilistic model, it is easy to make $\Psi(\theta)$ quite large and then assign very small probabilities to larger errors. The problem with nondeterministic uncertainty is that $\Psi(\theta)$ needs to be smaller to make appropriate decisions; however, theoretically "guaranteed" performance may not truly be guaranteed in practice.

Once a nondeterministic model is formulated, the optimal decision rule may produce results that seem absurd for the intended application. The problem is that the DM cannot tolerate any risk. An action is applied only if the result can be guaranteed. The hope of doing better than the worst case is not taken into account. Consider the following example:

Example 9.28 (A Problem with Conservative Decision Making) Suppose that a friend offers you the choice of either a check for 1000 Euros or 1 Euro in cash. With the check, you must take it to the bank, and there is a small chance that your friend will have insufficient funds in the account. In this case, you will receive nothing. If you select the 1 Euro in cash, then you are guaranteed to earn something.

The following cost matrix reflects the outcomes (ignoring utility theory):

$$\begin{array}{cc} & U \\ \Theta & \begin{array}{|c|c|} \hline 1 & 1000 \\ \hline 1 & 0 \\ \hline \end{array} \end{array} . \tag{9.93}$$

Using probabilistic analysis, we might conclude that it is best to take the check. Perhaps the friend is even known to be very wealthy and responsible with banking accounts. This information, however, cannot be taken into account in the decision-making process. Using worst-case analysis, the optimal action is to take the 1 Euro in cash. You may not feel too good about it, though. Imagine the regret if you later learn that the account had sufficient funds to cash the check for 1000 Euros. ■

Thus, it is important to remember the price that one must pay for wanting results that are absolutely guaranteed. The probabilistic model offers the flexibility of incorporating statistical information. Sometimes the probabilistic model can be viewed as a generalization of the nondeterministic model. If it is assumed that nature acts after the robot, then the nature action can take this into account, as incorporated into Formulation 9.4. In the nondeterministic case, $\Theta(u)$ is specified, and in the probabilistic case, $P(\theta|u)$ is specified. The distribution $P(\theta|u)$ can be designed so that nature selects with very high probability the $\theta \in \Theta$ that maximizes $L(u, \theta)$. In Example 9.28, this would mean that the probability that

the check would bounce (resulting in no earnings) would by very high, such as 0.999999. In this case, even the optimal action under the probabilistic model is to select the 1 Euro in cash. For virtually any decision problem that is modeled using worst-case analysis, it is possible to work backward and derive possible priors for which the same decision would be made using probabilistic analysis. In Example 9.4, it seemed as if the decision was based on assuming that with very high probability, the check would bounce, even though there were no probabilistic models.

This means that worst-case analysis under the nondeterministic model can be considered as a special case of a probabilistic model in which the prior distribution assigns high probabilities to the worst-case outcomes. The justification for this could be criticized in the same way that other prior assignments are criticized in Bayesian analysis. What is the basis of this particular assignment?

9.5.4 Concerns Regarding Game Theory

One of the most basic limitations of game theory is that each player must know the cost functions of the other players. As established in Section 9.5.1, it is even quite difficult to determine an appropriate cost function for a single decision maker. It is even more difficult to determine costs and motivations of other players. In most practical settings this information is not available. One possibility is to model uncertainty associated with knowledge of the cost function of another player. Bayesian analysis could be used to reason about the cost based on observations of actions chosen by the player. Issues of assigning priors once again arise. One of the greatest difficulties in allowing uncertainties in the cost functions is that a kind of "infinite reflection" occurs [392]. For example, if I am playing a game, does the other player know my cost function? I may be uncertain about this. Furthermore, does the other player know that I do not completely know its cost function? This kind of second-guessing can occur indefinitely, leading to a nightmare of nested reasoning and assignments of prior distributions.[10]

The existence of saddle points or Nash equilibria was assured by using randomized strategies. Mathematically, this appears to be a clean solution to a frustrating problem; however, it also represents a substantial change to the model. Many games are played just once. For the expected-case results to converge, the game must be played an infinite number of times. If a game is played once, or only a few times, then the players are very likely to experience regret, even though the theory based on expected-case analysis indicates that regret is eliminated.

Another issue is that intelligent human players may fundamentally alter their strategies after playing a game several times. It is very difficult for humans to simulate a randomized strategy (assuming they even want to, which is unlikely). There are even international tournaments in which the players repeatedly engage

[10]Readers familiar with the movie *The Princess Bride* may remember the humorous dialog between Vizzini and the Dread Pirate Roberts about which goblet contains the deadly Iocane powder.

in classic games such as Rock-Paper-Scissors or the Prisoner's Dilemma. For an interesting discussion of a tournament in which people designed programs that repeatedly compete on the Prisoner's Dilemma, see [917]. It was observed that even some cooperation often occurs after many iterations, which secures greater rewards for both players, even though they cannot communicate. A famous strategy arose in this context called Tit-for-Tat (written by Anatol Rapoport), which in each stage repeated the action chosen by the other player in the last stage. The approach is simple yet surprisingly successful.

In the case of nonzero-sum games, it is particularly disheartening that multiple Nash equilibria may exist. Suppose there is only one admissible equilibrium among several Nash equilibria. Does it really seem plausible that an adversary would think very carefully about the various Nash equilibria and pick the admissible one? Perhaps some players are conservative and even play security strategies, which completely destroys the assumptions of minimizing regret. If there are multiple admissible Nash equilibria, it appears that regret is unavoidable unless there is some collaboration between players. This result is unfortunate if such collaboration is impossible.

Further Reading

Section 9.1 covered very basic concepts, which can be found in numerous books and on the Internet. For more on Pareto optimality, see [847, 909, 953, 1005]. Section 9.2 is inspired mainly by decision theory books. An excellent introduction is [89]. Other sources include [268, 271, 673, 831]. The "game against nature" view is based mainly on [109]. Pattern classification, which is an important form of decision theory, is covered in [19, 271, 295, 711]. Bayesian networks [778] are a popular representation in artificial intelligence research and often provide compact encodings of information for complicated decision-making problems.

Further reading on the game theory concepts of Sections 9.3 and 9.4 can be found in many books (e.g., [59, 759]). A fun book that has many examples and intuitions is [917]. For games that have infinite action sets, see [59]. The computation of randomized Nash equilibria remains a topic of active research. A survey of methods appears in [691]; see also [545, 699]. The coupled polynomial equations that appear in computing randomized Nash equilibria may seem to suggest applying algorithms from computational algebraic geometry, as were needed in Section 6.4 to solve this kind of problem in combinatorial motion planning. An approach that uses such tools is given in [261]. Contrary to the noncooperative games defined in Section 9.4, *cooperative game theory* investigates ways in which various players can form coalitions to improve their rewards [779].

Parts of Section 9.5 were inspired by [89]. Utility theory appears in most decision theory books (e.g., [89]) and in some artificial intelligence books (e.g., [839]). An in-depth discussion of Bayesian vs. frequentist issues appears in [831]. For a thorough introduction to constructing cost models for decision making, see [539].

Exercises

1. Suppose that a single-stage two-objective decision-making problem is defined in which there are two objectives and a continuous set of actions, $U = [-10, 10]$. The cost vector is $L = [u^2 \;\; u - 1]$. Determine the set of Pareto-optimal actions.

2. Let

		Θ		
	-1	3	2	-1
U	-1	0	7	-1
	1	5	5	-2

   define the cost for each combination of choices by the decision maker and nature. Let nature's randomized strategy be $[1/5 \;\; 2/5 \;\; 1/10 \;\; 3/10]$.

   (a) Use nondeterministic reasoning to find the minimax decision and worst-case cost.

   (b) Use probabilistic reasoning to find the best expected-case decision and expected cost.

3. Many reasonable decision rules are possible, other than those considered in this chapter.

   (a) Exercise 2(a) reflects extreme pessimism. Suppose instead that extreme optimism is used. Select the choice that optimizes the best-case cost for the matrix in Exercise 2.

   (b) One approach is to develop a coefficient of optimism, $\alpha \in [0, 1]$, which allows one to interpolate between the two extreme scenarios. Thus, a decision, $u \in U$, is chosen by minimizing

$$\alpha \max_{\theta \in \Theta} \Big\{ L(u, \theta) \Big\} + (1 - \alpha) \min_{\theta \in \Theta} \Big\{ L(u, \theta) \Big\}. \tag{9.94}$$

   Determine the optimal decision for this scenario under all possible choices for $\alpha \in [0, 1]$. Give your answer as a list of choices, each with a specified range of α.

4. Suppose that after making a decision, you observe the choice made by nature. How does the cost that you received compare with the best cost that could have been obtained if you chose something else, given this choice by nature? This difference in costs can be considered as *regret* or minimum "Doh!"[11] Psychologists have argued that some people make choices based on minimizing regret. It reflects how badly you wish you had done something else after making the decision.

   (a) Develop an expression for the worst-case regret, and use it to make a minimax regret decision using the matrix from Exercise 2.

[11] In 2001, the Homer Simpson term "Doh!" was added to the Oxford English Dictionary as an expression of regret.

(b) Develop an expression for the expected regret, and use it to make a minimum expected regret decision.

5. Using the matrix from Exercise 2, consider the set of all probability distributions for nature. Characterize the set of all distributions for which the minimax decision and the best expected decision results in the same choice. This indicates how to provide reverse justification for priors.

6. Consider a Bayesian decision-theory scenario with cost function L. Show that the decision rule never changes if $L(u, \theta)$ is replaced by $aL(u, \theta) + b$, for any $a > 0$ and $b \in \mathbb{R}$.

7. Suppose that there are two classes, $\Omega = \{\omega_1, \omega_2\}$, with $P(\omega_1) = P(\omega_2) = \frac{1}{2}$. The observation space, Y, is $\mathbb{R}$. Recall from probability theory that the normal (or Gaussian) probability density function is

$$p(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(y-\mu)^2/2\sigma^2}, \tag{9.95}$$

in which μ denotes the mean and σ^2 denotes the variance. Suppose that $p(y|\omega_1)$ is a normal density in which $\mu = 0$ and $\sigma^2 = 1$. Suppose that $p(y|\omega_2)$ is a normal density in which $\mu = 6$ and $\sigma^2 = 4$. Find the optimal classification rule, $\gamma : Y \to \Omega$. You are welcome to solve the problem numerically (by computer) or graphically (by careful function plotting). Carefully explain how you arrived at the answer in any case.

8. Let

		Θ		
	2	-2	-2	1
U	-1	-2	-2	6
	4	0	-3	4

give the cost for each combination of choices by the decision maker and nature. Let nature's randomized strategy be [1/4 1/2 1/8 1/8].

(a) Use nondeterministic reasoning to find the minimax decision and worst-case cost.

(b) Use probabilistic reasoning to find the best expected-case decision and expected cost.

(c) Characterize the set of all probability distributions for which the minimax decision and the best expected decision results in the same choice.

9. In a *constant-sum game*, the costs for any $u \in U$ and $v \in V$ add to yield

$$L_1(u, v) + L_2(u, v) = c \tag{9.96}$$

for some constant c that is independent of u and v. Show that any constant-sum game can be transformed into a zero-sum game, and that saddle point solutions can be found using techniques for the zero-sum formulation.

10. Formalize Example 9.7 as a zero-sum game, and compute security strategies for the players. What is the expected value of the game?

11. Suppose that for two zero-sum games, there exists some nonzero $c \in \mathbb{R}$ for which the cost matrix of one game is obtained by multiplying all entries by c in the cost matrix of the other. Prove that these two games must have the same deterministic and randomized saddle points.

12. In the same spirit as Exercise 11, prove that two zero-sum games have the same deterministic and randomized saddle points if c is added to all matrix entries.

13. Prove that multiple Nash equilibria of a nonzero-sum game specified by matrices A and B are interchangeable if (A, B) as a game yields the same Nash equilibria as the game $(A, -A)$.

14. Analyze the game of Rock-Paper-Scissors for three players. For each player, assign a cost of 1 for losing, 0 for a tie, and -1 for winning. Specify the cost functions. Is it possible to avoid regret? Does it have a deterministic Nash equilibrium? Can you find a randomized Nash equilibrium?

15. Compute the randomized equilibrium point for the following zero-sum game:

$$\begin{array}{cc} & V \\ U & \begin{array}{|c|c|} \hline 0 & \text{-1} \\ \hline \text{-1} & 2 \\ \hline \end{array} \end{array} . \tag{9.97}$$

    Indicate the randomized strategies for the players and the resulting expected value of the game.

Implementations

16. Consider estimating the value of an unknown parameter, $\theta \in \mathbb{R}$. The prior probability density is a normal,

$$p(\theta) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(\theta-\mu)^2/2\sigma^2}, \tag{9.98}$$

    with $\mu = 0$ and $\sigma = 4$. Suppose that a sequence, y_1, y_2, ..., y_k, of k observations is made and that each $p(y_i|\theta)$ is a normal density with $\mu = \theta$ and $\sigma = 9$. Suppose that u represents your guess of the parameter value. The task is select u to minimize the expectation of the cost, $L(u, \theta) = (u - \theta)^2$. Suppose that the "true" value of θ is 4. Determine the u^*, the minimal action with respect to the expected cost after observing: $y_i = 4$ for every $i \in \{1, \ldots, k\}$.

    (a) Determine u^* for $k = 1$.

    (b) Determine u^* for $k = 10$.

    (c) Determine u^* for $k = 1000$.

This experiment is not very realistic because the observations should be generated by sampling from the normal density, $p(y_i|\theta)$. Repeat the exercise using values drawn with the normal density, instead of $y_k = 4$, for each k.

17. Implement an algorithm that computes a randomized saddle point for zero-sum games. Assume that one player has no more than two actions and the other may have any finite number of actions.

18. Suppose that a K-stage decision-making problem is defined using multiple objectives. There is a finite state space X and a finite action set $U(x)$ for each $x \in X$. A state transition equation, $x_{k+1} = f(x_k, u_k)$, gives the next state from a current state and input. There are N cost functionals of the form

$$L_i(u_1, \ldots, u_K) = \sum_{k=1}^{K} l(x_k, u_k) + l_F(x_F), \tag{9.99}$$

in which $F = K + 1$. Assume that $l_F(x_F) = \infty$ if $x_F \in X_{goal}$ (for some goal region $X_{goal} \subset X$) and $l_F(x_F) = 0$ otherwise. Assume that there is no termination action (which simplifies the problem). Develop a value-iteration approach that finds the complete set of Pareto-optimal plans efficiently as possible. If two or more plans produce the same cost vector, then only one representative needs to be returned.