REINFORCEMENT LEARNING AND STOCHASTIC OPTIMIZATION

A unified framework for sequential decisions

Warren B. Powell

August 22, 2021

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Optimization Under Uncertainty: A unified framework
Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

CHAPTER 10

UNCERTAINTY MODELING

We cannot find an effective policy unless we are modeling the problem properly. In the realm of sequential decision problems, this means accurately modeling uncertainty. The importance of modeling uncertainty has been underrepresented in the stochastic optimization literature, although practitioners working on real problems have long been aware of both the importance and the challenges of modeling uncertainty.

Fortunately, there is a substantial body of research focused on the modeling of uncertainty and stochastic processes that has evolved in the communities working on Monte Carlo simulation and uncertainty quantification. We use uncertainty modeling as the broader term that describes the process of identifying and modeling uncertainty, while simulation refers to the vast array of tools that break down complex stochastic processes using the computational tools of Monte Carlo simulation.

It helps to remind ourselves of the two information processes that drive any sequential stochastic optimization problem: decisions, and exogenous information. Assume that we can pick some policy $X_t^\pi(S_t)$. We need to be able to simulate a sample realization of the policy, which will look like

$$S_0 \rightarrow x_0 = X_0^\pi(S_0) \rightarrow W_1 \rightarrow S_1 \rightarrow x_1 = X_1^\pi(S_1) \rightarrow W_2 \rightarrow S_3 \rightarrow$$

Given our policy, this simulation assumes that we have access to a transition function

$$S_{t+1} = S^M(S_t, X_t^\pi(S_t), W_{t+1}). \tag{10.1}$$

We can execute equation (10.1) if we are given a policy $X_t^\pi(S_t)$ and if we have access to the following:

S_0 = The initial state - This is where we place information about initial estimates (or priors) of parameters, as well as assumptions about probability distributions and functions.

W_t = Exogenous information that enters our system for the first time between $t-1$ and t for $t = 1, 2, \ldots, T$.

In this chapter, we focus on the often challenging problem of simulating the exogenous sequence $(W_t)_{t=0}^T$. We assume that the initial state S_0 is given, but recognize that it may include a probabilistic belief about unknown and unobservable parameters. The process of converting the characteristics of a stochastic process into a mathematical model is broadly known as *uncertainty quantification.* Since it is easy to overlook sources of uncertainty when building a model, we place considerable attention on identifying the different sources of uncertainty that we have encountered in our applied work, keeping in mind that S_0 and W_t are the only variables our modeling framework provides for representing uncertainty.

After reviewing different sources of uncertainty, we then provide a basic introduction to a powerful set of techniques known as Monte Carlo simulation, which allows us to replicate stochastic processes on the computer. Given the rich array of different types of stochastic processes, our discussion here provides little more than a taste of the tools that are available to replicate stochastic processes.

10.1 SOURCES OF UNCERTAINTY

Uncertainty arises in different forms. Some of the major forms that we have encountered are

- Observational errors - This arises from uncertainty in observing or measuring the state of the system. Observational errors arise when we have unknown state variables that cannot be observed directly (and accurately).

- Exogenous uncertainty - This describes the exogenous arrival of information to the system, which might be weather, demands, prices, the response of a patient to medication or the reaction of the market to a product.

- Prognostic uncertainty - We often have access to a forecast $f_{tt'}^W$ of the information $W_{t'}$. Prognostic uncertainty captures the deviation of the actual $W_{t'}$ from the forecast $f_{tt'}^W$. If we think of $W_t = f_{tt}^W$ as the actual value of W_t, then we can think of the realization of W_t (the exogenous information described above) as just an update to a forecast.

- Inferential (or diagnostic) uncertainty - Inferential uncertainty arises when we use observations (from field or physical measurements, or computer simulations) to draw inferences about another set of parameters. It arises from our lack of understanding of the precise properties or behavior of a system, which introduces errors in our ability to estimate parameters, partly from noise in the observations, and partly from errors in our modeling of the underlying system.

- Experimental variability - Sometimes equated with observational uncertainty, experimental variability refers to differences between the results of experiments run under similar conditions. An experiment might be a computer simulation, a laboratory experiment or a field implementation. Even if we can perfectly measure the results of an experiment, there is variation from one experiment to the next.

- Model uncertainty - We may not know the structure of the transition function $S_{t+1} = S^M(S_t, x_t, W_{t+1})$, or the parameters that are imbedded in the function. Model uncertainty is often attributed to the transition function, but it may also apply to the model of the stochastic process W_t since we often do not know the precise structure.

- Transitional uncertainty - This arises when we have a perfect model of how a system should evolve, but exogenous shocks (wind buffeting an aircraft, rainfall affecting reservoir levels) can introduce uncertainty in how an otherwise deterministic system will evolve. Transitional uncertainty is often represented as

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

- Control/implementation uncertainty - This is where we choose a control u_t (such as a temperature or speed), but what happens is $\hat{u}_t = u_t + \delta u_t$ where δu_t is a random perturbation.

- Communication errors and biases - Communication from an agent q about his state S_{qt} to an agent q' where errors may introduced, either accidentally or purposely.

- Algorithmic instability - Very minor changes in the input data for a problem, or small adjustments in parameters guiding an algorithm (which exist in virtually all algorithms), can completely change the path of the algorithm, introducing variability in the results.

- Goal uncertainty - Uncertainty in the desired goal of a solution, as might arise when a single model has to produce results acceptable to different people or users.

- Political/regulatory uncertainty - Uncertainty about taxes, rules and requirements that affect costs and constraints (for example, tax energy credits, automotive mileage standards). These can be viewed as a form of systematic uncertainty, but this is a particularly important source of uncertainty with its own behaviors.

Below we provide more detailed discussions of each type of uncertainty. One challenge is modeling each source of uncertainty, since we have only two mechanisms for introducing exogenous information into our model: the initial state S_0, and the exogenous information process $W_1, W_2, \ldots$. Thus, the different types of uncertainty may look similar mathematically, but it is important to characterize the mechanisms by which uncertainty enters our model.

10.1.1 Observational errors

Observational (or measurement) uncertainty reflects errors in our ability to observe (or measure) the state of the system directly. Some examples include:

EXAMPLE 10.1

Different people may measure the gases in the oil of a high-voltage transformer, producing different measurements (possibly due to variations in equipment, the temperature at which the transformer was observed, or variations in the oil surrounding the coils).

EXAMPLE 10.2

The Center for Disease Control and Prevention estimates the number of mosquitoes carrying a disease by setting traps and counting how many mosquitoes are caught that are found with the disease. From day to day the number of infected mosquitoes that are caught can vary considerably.

EXAMPLE 10.3

A company may be selling a product at a price p_t which is being varied to find the best price. However, the sales (at a fixed price) will be random from one time period to the next.

EXAMPLE 10.4

Different doctors, seeing the same patient for the first time, may elicit different information about the characteristics of the patient.

Partially observable systems arise in any application where we cannot directly observe parameters. A simple example arises in pricing, where we may feel that demand varies linearly with price according to

$$D(p) = \theta_0 - \theta_1 p.$$

At time t, our best estimate of the demand function is given by

$$D(p) = \bar{\theta}_0 - \bar{\theta}_1 p.$$

We observe sales, which would be given by

$$\hat{D}_{t+1} = \theta_0 - \theta_1 p_t + \varepsilon_{t+1}.$$

We do not know (θ_0, θ_1), but we can use observations to create updated estimates. If $(\bar{\theta}_{t0}, \bar{\theta}_{t1})$ is our estimate as of time t, we can use our observation $\hat{D}_{t+1}$ of sales between t and $t+1$ to obtain updated estimates $(\bar{\theta}_{t+1,0}, \bar{\theta}_{t+1,1})$. In this model, we would view $\bar{\theta}_t = (\bar{\theta}_{t0}, \bar{\theta}_{t1})$ as our state variable, which is our estimate of the static parameter θ. Since θ is a fixed parameter, we do not include it in the state variable, but rather treat it as a *latent variable*.

The presence of states that cannot be perfectly observable gives rise to what are widely known as *partially observable Markov decision processes*, or POMDP's. To model this, let $\check{S}_t$ be the true (but possibly unobservable) state of the system at time t, while S_t is the observable state. One way of writing our dynamics might be

$$S_{t+1} = \check{S}^M(\check{S}_t, x_t) + \varepsilon_{t+1},$$

which captures our inability to directly observe $\check{S}_t$. These systems are most often motivated by problems such as those in engineering where we cannot directly observe the state of charge of a battery, the location and velocity of an aircraft, or the number of truck trailers sitting at a terminal (terminal managers tend to hide trailers to keep up their inventories).

We can represent our unobservable state as a probability distribution. This might be a continuous distribution (perhaps the normal or multivariate normal distribution), or perhaps more simply as a discrete distribution where q^k_{ti} is the probability that the state variable S_{ti} takes on outcome k (or perhaps a parameter θ^k) at time t. Then, the vector $q_{ti} = (q^k_{ti}), k = 1, \ldots, K$ is the distribution capturing our belief about the unobservable state. We then include q_t (for each uncertain state dimension) as part of our state variable (this is where our belief state comes in).

10.1.2 Exogenous uncertainty

Exogenous uncertainty represents the information that we typically model through the process W_t represent new information about supplies and demands, costs and prices, and physical parameters that can appear in either the objective function or constraints. Exogenous uncertainty can arise in different styles, including:

- Fine-grained time-scale uncertainty - Sometimes referred to as *aleatoric uncertainty*, fine time-scale uncertainty refers to uncertainty that varies from time-step to time-step which is assumed to reflect the dynamics of the problem. Whether a time step is minutes, hours, days or weeks, fine time-scale uncertainty means that information from one time-step to the next is either uncorrelated, or where correlations drop off fairly quickly.

- Coarse-grained time-scale uncertainty - Referred to in different settings as systematic uncertainty or *epistemic* uncertainty (popular in the medical community), coarse time-scale uncertainty reflects uncertainty in an environment which occurs over long time scales. This might reflects new technology, changes in market patterns, the introduction of a new disease, or an unobserved fault in machinery for a process.

- Distributional uncertainty - If we represent the exogenous information W_t, or the initial state S_0, as a probability distribution, there may be uncertainty in either the type of distribution or the parameters of a distribution.

- Adversarial uncertainty - The exogenous information process $W_1, \ldots, W_T$ may come from another agent who is choosing W_t in a way to make us perform poorly. We cannot be sure how the adversary may behave.

10.1.3 Prognostic uncertainty

Prognostic uncertainty reflects errors in our ability to forecast activities in the future. Typically these are written as $f_{tt'}$ to represent the forecast of some quantity at time t', given what we know at time t (represented by our state variable S_t). Examples include:

EXAMPLE 10.5

A company may create a forecast of demand D_t for its product. If $f^D_{tt'}$ is the forecast of the demand $D_{t'}$ given what we know at time t, then the difference between $f^D_{tt'}$ and $D_{t'}$ is the uncertainty in our forecast.

EXAMPLE 10.6

A utility is interested in forecasting the price of electricity 10 years from now. Electricity prices are well approximated by the intersection of the load (the amount of electricity needed at a point in time) and the "supply stack" which is the cost of energy as a function of the total supply (typically an increasing function). The supply stack reflects the cost of different fuels (nuclear, coal, natural gas) and generators (different technologies, and different ages, affect operating costs). We have to forecast the prices of these different sources (one form of uncertainty) along with the load (a different form of uncertainty).

EXAMPLE 10.7

We might be interested in forecasting energy from wind $E^W_{t'}$ at time t. This might require that we first generate a meteorological forecast of weather systems (high and low pressure systems), as well as capturing the movement of the atmosphere (wind speed and direction).

If $W_{t'}$ is some form of random information in the future, we might be able to create a forecast $f^W_{tt'}$ using what we know at time t. We typically assume that our forecasts are unbiased, which means we can write

$$f^W_{tt'} = \mathbb{E}\{W_{t'}|S_t\}.$$

Forecasts can come from two sources. An *endogenous forecast* is obtained from a model that is created endogenously from data. For example, we might be forecasting demand using the model

$$f^D_{tt'} = \theta_{t0} + \theta_{t1}(t' - t).$$

Now assume we observe the demand D_{t+1}. We might use any of a range of algorithms to update our parameter estimates to obtain

$$f^D_{t+1,t'} = \theta_{t+1,0} + \theta_{t+1,1}(t' - (t+1)).$$

The parameter vector θ_t can be updated recursively from observations W_{t+1}. If θ_t is our current estimate of $(\theta_{t0}, \theta_{t1})$, let Σ_t be our estimate of the covariance between the random variables θ_0 and θ_1 (these are the true values of the parameters). Let $\beta^W = 1/(\sigma^2_W)$ be the precision of an observation W_{t+1} (the precision is the inverse of the variance), and assume we can form the precision matrix given by $M_t = [(X_t)^T X_t]^{-1}$, where X_t is a matrix where each row consists of the vector of independent variables. In the case of our demand example, the design variables for time t would be $x_t = (1 \;\; p_t)$. We can update θ_t and Σ_t (or M_t) recursively using

$$\theta_{t+1} = \theta_t - \frac{1}{\gamma_{t+1}} M_t x_{t+1} \varepsilon_{t+1}, \tag{10.2}$$

where ε_{t+1} is the error given by

$$\varepsilon_{t+1} = W_{t+1} - \theta_t x_t. \tag{10.3}$$

The matrix $M_{t+1} = [(X_{t+1})^T X_{t+1}]^{-1}$. This can be updated recursively without computing an explicit inverse using

$$M_{t+1} = M_t - \frac{1}{\gamma_{t+1}} (M_t x_{t+1} (x_{t+1})^T M_t). \tag{10.4}$$

The parameter γ_{t+1} is a scalar computed using

$$\gamma_{t+1} \quad = \quad 1 + (x_{t+1})^T M_t x_{t+1}. \tag{10.5}$$

Note that if we multiply (10.4) through by σ_ϵ^2 we obtain

$$\Sigma_{t+1}^\theta = \Sigma_t^\theta - \frac{1}{\gamma_{t+1}} (\Sigma_t^\theta x_{t+1} (x_{t+1})^T \Sigma_t^\theta), \tag{10.6}$$

where we scale γ_{t+1} by σ_ϵ^2, giving us

$$\gamma_{t+1} \quad = \quad \sigma_\epsilon^2 + (x_{t+1})^T \Sigma_t^\theta x_{t+1}. \tag{10.7}$$

Equations (10.2)-(10.7) represent the transition function for updating θ_t.

The second source of a forecast is exogenous, where the forecast might be supplied by a vendor. In this case, we might view the updated set of forecasts $(f_{tt'})_{t' \geq t}$ as exogenous information. Alternatively, we could think of the change in forecasts as the exogenous information. If we let $\hat{f}_{t+1,t'}$ be the change between t and $t+1$ in the forecast for activities at time t', we would then write

$$f_{t+1,t'} = f_{tt'} + \hat{f}_{t+1,t'}.$$

From a modeling perspective, these forecasts differ in terms of how they are represented in the state variable. In the case of our endogenous forecast, the state variable would be captured by (θ_t, Σ_t), with the corresponding transition equations given by (10.2)-(10.7). With our exogenous forecast, the state variable would be simply $(f_{tt'})_{t'=t}^T$.

Regardless of whether the forecast is exogenous or endogenous, the new information (the exogenous observation or the updated forecast) would be modeled as a part of the exogenous information process W_t.

10.1.4 Inferential (or diagnostic) uncertainty

It is often the case that we cannot directly observe a parameter. Instead, we have to use (possibly imperfect) observations of one or more parameters to infer variables or parameters that we cannot directly observe. Some examples include:

■ **EXAMPLE 10.8**

We might not be able to directly observe the presence of heart disease, but we may use blood pressure as an indicator. Measuring blood pressure introduces observational error, but there is also error in making the inference that a patient suffers from heart disease from blood pressure alone.

■ **EXAMPLE 10.9**

We observe (possibly imperfectly) the sales of a product. From these sales we wish to estimate the elasticity of demand with respect to price.

■ **EXAMPLE 10.10**

Power companies generally do not know the precise location of a tree falling that creates a power outage. Rather, a falling tree can create a short circuit that will trip a circuit breaker higher in the tree (rooted at a substation), producing many outages, including to customers who may be far from the fallen tree. Diagnostic uncertainty refers to errors in our ability to precisely describe where a tree might have fallen purely from phone calls.

■ **EXAMPLE 10.11**

Sensors may detect an increase in carbon monoxide in the exhaust of a car. This information may indicate several possible causes, such as an aging catalytic converter, the improper timing of the cylinders, or an incorrect air-fuel mixture (which might hint at a problem in a different sensor).

Inferential uncertainty can be described as uncertainty in the parameters of a model. In our example involving the detection of carbon monoxide, we might use this information to update the probability that the real cause is due to each of three or four different mechanical problems. This would represent an instance of using a (possibly noisy) observation to update a lookup table model of where failures are located. By contrast, when we use sales data to update our demand elasticity, that would be an example of using noisy observational data to update a parametric model.

In some settings the term *diagnostic uncertainty* is used instead of inferential uncertainty. We feel that this term reflects the context of identifying a problem (a failed component, presence of a disease) that we are not able to observe directly. However, both inferential uncertainty and diagnostic uncertainty reflect uncertainty in parameters that have been estimated (inferred) from indirect observations.

Inferential uncertainty is a form of derived uncertainty that arises when we estimate a parameter $\bar{\theta}$ from data (simulated or observed). The raw uncertainty is contained in the sequence W_t (or W^n). We then have to derive the distribution of our estimate $\bar{\theta}$ resulting from the exogenous noise, which we contain in our belief state B_t.

10.1.5 Experimental variability

Experimental variability reflects changes in the results of experiments run under the same conditions. Experimental settings include

Laboratory experiments - We include here physical experiments run in a laboratory setting, encompassing chemical, biological, mechanical and even human testing.

Numerical simulations - Large simulators describing complex physical systems, ranging from models of businesses to models of physical processes, can exhibit variability from one run to the next, often reflecting minor variations in input data and parameters.

Field testing - This can range from observing sales of a product to testing of new drugs.

Experimental uncertainty arises from possibly minor variations in the dynamics of a system (simulated or physical) which introduce variability when running experiments. Experimental uncertainty typically reflects our inability to perfectly estimate parameters that drive the system, or errors in our ability to understand (or model) the system.

Some sources equate observational and experimental uncertainty, and often they are handled in the same way. However, we feel it is useful to distinguish between pure measurement (observational) errors, which might be reduced with better technology, and experimental errors, which have more to do with the process and which are not reduced through better measurement technologies.

Experimental noise might be attributed as a byproduct of the exogenous information process W_t. For example, for a given policy $X^\pi(S_t)$, an experiment might consist of evaluating

$$\hat{F}^\pi = F^\pi(\omega) = \sum_{t=0}^{T} C(S_t(\omega), X^\pi(S_t(\omega))).$$

Here, the noise is due to the variation in W_t. However, imagine that we are running a series of experiments. Let $\hat{F}^n(\theta^n)$ be the observation of the outcome of an experiment run with parameters $\theta = \theta^n$. Let $f(\theta) = \mathbb{E}\hat{F}^n(\theta)$ be the exact (but unobservable) value of running the experiment with parameter setting θ. We can write

$$\hat{F}^n(\theta^n) = f(\theta^n) + \varepsilon^n.$$

In this case, the sequence ε^n would be the exogenous information information W^n.

10.1.6 Model uncertainty

"Model uncertainty" is a bit of a catch-all phrase that often refers to the transition function, but not always. Model uncertainty comes in two forms. The first is errors in estimates of parameters of a parametric model. If we are estimating these parameters over time from observations, we would refer to this as inferential uncertainty. Now imagine that we characterize our model using a set of fixed parameters that are not being updated. We are not estimating these parameters over time, but rather we are just using assumed values which are uncertain.

The second is errors in the structure of the model itself (economists refer to this as *specification errors*). Some examples include:

EXAMPLE 10.12

We may approximate demand as a function of price as a linear function, a logistics curve, or a quadratic function. We will use observational data to estimate the parameters of each function, but we may not directly address the errors introduced by assuming a particular type of function.

EXAMPLE 10.13

We may describe the diffusion of chemicals in a liquid using a first-order set of differential equations, which we fit to observational data. However, the real process may be better described by a second (or higher) order set of differential equations. Our first-order model may be at best a good local approximation.

EXAMPLE 10.14

Grid operators often model the supply curve of a power generator using a convex function, which is easier to solve. However, a more detailed model might capture complex relationships that reflect the fact that costs may rise in steps as different components of the generator come on (e.g. heat recovery).

Model uncertainty for dynamic problems can be found in four different parts of the model:

- Costs or rewards - Measuring the cost of a grid outage on the community may require estimating the impact of a loss of power on homes and businesses.
- Constraints - Constraints can often be written in the form $A_t x_t = R_t$. There are many applications where dynamic uncertainty enters through the right hand side R_t; this is how we would model the supply or demand of blood which would be a more typical form of dynamic uncertainty. Model uncertainty often arises in the matrix A_t, which is where we might capture the assumed speed of an aircraft, or the efficiency of a manufacturing process.
- Stochastic modeling - If we are using a model of the exogenous information W_t, then there may be errors in this model.
- Dynamics - This is where we are uncertain about the function $S^M(S_t, x_t, W_{t+1})$ which describes how the system evolves over time.

The transition function $S^M(\cdot)$ captures all the physics of a problem, and there are many problems where we simply do not understand the physics. For example, we might be trying to explain how a person or market might respond to a price, or how global warming might respond to a change in CO2 concentrations.

Some policies make decisions using nothing more than the current state, allowing them to be used in settings where the underlying dynamics have not been modeled. By contrast, an entire class of policies based on lookahead models (which we cover in chapter 19) depend on at least an approximate model of the problem. See 9.7.2 for a discussion of model-free dynamic programming.

Whether we are dealing with costs, constraints or the dynamics, our model can be described in terms of the choice of the model structure, and any parameters that characterize the model. Let $m \in \mathcal{M}$ represent the structure of the model, and let $\theta \in \Theta^m$ be the parameters that characterize a model with structure m. As a general rule, the model structure m is fixed in advance (for example, we might assume that a particular relationship is linear) but with uncertain parameters.

An alternative approach is to associate a prior q_0^m that gives the probability that we believe that model m is correct. Similarly, we might start with an initial estimate θ_0^m for the parameter vector θ^m. We might even assume that we start by assuming that θ^m is described by a multivariate normal distribution with mean θ_0^m and covariance matrix Σ_0^m.

As we might expect, prior information about the model (whether it is the probability q_0 that a type of model is correct, or the prior distribution on θ^m) is communicated through the initial state S_0. If this belief is updated over time, then this would also be part of the dynamic state S_t.

10.1.7 Transitional uncertainty

There are many problems where the dynamics of the system are modeled deterministically. This is often the case in engineering applications where we apply a control u_t (such as a force) to a dynamic system. Simple physics might describe how the control affects our system, which we would then write

$$S_{t+1} = S^M(S_t, u_t).$$

However, exogenous noise might interfere with these dynamics. For example, we might be predicting the speed and location of an aircraft after applying forces u_t. Variations in the atmosphere might interfere with our equations, so we introduce a noise term ε_{t+1}, giving us

$$S_{t+1} = S^M(S_t, u_t) + \varepsilon_{t+1}.$$

We note that despite the noise, we assume that we can observe (measure) the state perfectly.

10.1.8 Control/implementation uncertainty

There are many problems where we cannot precisely control a process. Some examples include:

EXAMPLE 10.15

An experimentalist has requested that a rat be fed a diet with x_t grams of fat. However, variability in the preparation of the meals, and the choice the rat makes of what to eat, introduces variability in the amount of fat that is consumed.

EXAMPLE 10.16

A publisher chooses to sell a book at a wholesale price p_t^W at time t and then observes sales. However, the publisher has no control over the retail price offered to the purchasing public.

■ **EXAMPLE 10.17**

The operator of a power grid may request that a generator come online and generate x_t megawatts of power. However, this may not happen either because of a technical malfunction or human implementation errors.

Control uncertainty is widely overlooked in the dynamic programming literature, but is well known in the econometrics community as the "errors in variable" model.

We might model errors in the implementation of a decision using a simple additive model

$$\hat{x}_t = x_t + \varepsilon_t^x,$$

where $\hat{x}_t$ is the decision that is actually implemented, and ε_t^x captures the difference between what was requested, x_t, versus what was implemented, $\hat{x}_t$. We note that ε_t^x would be modeled as an element of W_t, although in practice it is not always observable.

It is important to distinguish between uncertainty in how a decision (or control) is implemented from other sources of uncertainty because of potential nonlinearities in how the decision affects the results.

10.1.9 Communication errors and biases

In a multiagent system, one agent might communicate location or status to another agent, but this information can contain errors (a drone might not know its exact location) or biases (a fleet driver might report being on the road for fewer hours in order to be allowed to drive longer). In supply chain management, an engine manufacturer may send inflated production targets to suppliers to encourage suppliers to have enough inventory to handle problems, say, in the quality of parts that require more returns.

10.1.10 Algorithmic instability

A more subtle form of uncertainty is one that we refer to as algorithmic uncertainty. We use this category to describe uncertainty that is introduced by the algorithm used to solve a problem, which may also be partly attributable to the model itself. Three examples of how algorithmic uncertainty arises are

- Algorithms that depend on Monte Carlo sampling.
- Algorithms that exhibit sensitivity to small changes in the input data.
- Algorithms that produce different results even when run on exactly the same data, possibly due to variations in run times for a parallel implementation of an algorithm.
- Optimization algorithms for nonconvex problems where the optimal solution is highly dependent on the starting point(s), which may be randomly generated.

The stochastic gradient algorithm introduced in chapter 5, which we write using

$$x^{n+1} = x^n + \alpha_n \nabla_x F(x^n, W^{n+1}),$$

is a nice example of an algorithm that depends on Monte Carlo sampling, which is how we generated the observation W^{n+1}. These algorithms depend on carefully tuned stepsize policies for α_n to mitigate the effects of the noise.

The second type of algorithmic uncertainty arises due to the sensitivity that many deterministic optimization algorithms exhibit (in particular integer programs and nonlinear programs). Small changes in the input data can produce wide swings in the solution, although often there may be little or no change in the objective function. Thus, we may solve an optimization problem (perhaps this might be a linear program) that depends on a parameter θ. Let $F(\theta)$ be the optimal objective function and let $x(\theta)$ be the optimal solution. Small changes in θ can produce large (and unpredictable) changes in $x(\theta)$, which introduces a very real form of uncertainty.

The third type of uncertainty arises primarily with complex problems such as large integer programs that might take advantage of parallel processing. The behavior of these algorithms depends on the performance of the parallel processors, which can be affected by the presence of other jobs on the system. As a result, we can observe variability in the results, even when applied to exactly the same problem with the same data.

Algorithmic uncertainty is in the same class as experimental uncertainty, thus we defer to the discussion there for a description of how to model it.

10.1.11 Goal uncertainty

Many problems involve balancing multiple, competing objectives, such as putting different priorities on cost versus service, profits versus risk. One way to model this is to assume a linear utility function of the form

$$U(S,x) = \sum_{\ell \in \mathcal{L}} \theta_\ell \phi_\ell(S,x),$$

where S is our state variable, x is a decision, and $(\phi_\ell(S,x))_{\ell \in \mathcal{L}}$ is a set of features that capture the different metrics we use to evaluate a system such as cost, service, productivity, and total profits. The vector $(\theta_\ell)_{\ell \in \mathcal{L}}$ captures the weight we put on each feature. One way to model goal uncertainty is to represent θ as being uncertain (it may even vary from one decision-maker to another).

Another form of uncertainty might arise when we do not know all the features $\phi(S,x)$. For example, we may not even be aware that a reason to assign a particular driver to move a customer is that the customer is going to a location near the home of the driver. A human dispatcher might know this through personal interactions with the driver, but a computer might not. The result could then be a disagreement between a computer recommendation and what a human wants to do.

10.1.12 Political/regulatory uncertainty

For problems that involve long-term planning, changes in laws and regulations can introduce a significant source of uncertainty. Supply chain relationships with China, for example, can introduce the dimension of changes in tariffs. Planning energy investments bring in the dimension of the potential of a carbon tax. Manpower planning in many countries can depend on immigration policies, in industries ranging from agriculture to software to manufacturing.

10.1.13 Discussion

Careful readers will notice some overlap between these different types of uncertainty. Observational uncertainty, which refers specifically to errors in the direct observation of a parameter, and inferential uncertainty, which refers to errors in our ability to make inferences about models and parameters indirectly from data, represents one example, but we feel that it is useful to highlight the distinction. Model uncertainty is a term that resonates with many people, but it is an umbrella for several types of uncertainty.

What matters is if this list helps people identify as many sources of uncertainty as possible. We test this idea in a brief case study next.

10.2 A MODELING CASE STUDY: THE COVID PANDEMIC

A particularly rich application for modeling uncertainty arises when planning the vaccination response to the COVID pandemic, which was unfolding as this book was being written. Table 10.1 lists each of the different sources of uncertainty, and provides a few examples of each.

For a problem as complex as planning the vaccination process for COVID, there are many sources of uncertainty. Working from our list of different types of uncertainty helps to highlight forms of uncertainty that might be overlooked. Keep in mind that any model of a complex problem requires simplifications, but it helps to list as many sources of uncertainty as possible so that any simplifications are conscious ones, as opposed to simply overlooking a source of uncertainty.

10.3 STOCHASTIC MODELING

Once we have identified sources of uncertainty, the next step is generating sequences of random outcomes that represent samples of observations of exogenous information. This exercise can be relatively straightforward, or not. There are many problems where the stochastic modeling of the different sources of uncertainty is much harder, and much more important, than design a policy.

10.3.1 Sampling exogenous information

Somewhere in stochastic modeling we usually end up needing to compute an expectation, as we found in chapter 9 when we formulated our objective function as

$$\min_{\pi} \mathbb{E} \sum_{t=0}^{T} C(S_t, X_t^{\pi}(S_t)).$$

With rare exceptions, we will not be able to compute the expectation, and instead we have to resort to sampling, which can be accomplished in one of several ways:

- Mathematical models - Here we develop probability distributions to describe the frequency of different outcomes. We then use the methods of Monte Carlo simulation (described below) to sample from these distributions. This approach requires the highest mathematical sophistication to generate samples that mimic actual behavior.

Type of uncertainty	Description
Observational errors	Sample error observing people with symptoms Errors classifying people with symptoms as having COVID
Exogenous uncertainty	Reports of new cases, deaths Availability of ICUs, personal protective equipment Actual production of vaccines
Prognostic uncertainty	Projection of cases, hospital admissions Estimates of future performance of vaccines Projections of population response to vaccines Projections of vaccine production
Inferential uncertainty	Estimates of infection rates Estimates of effectiveness of vaccines
Experimental uncertainty	Uncertainty in how a drug will perform in a clinical trial Uncertainty in how many people will agree to be vaccinated
Model uncertainty	Uncertainty in the structure of the transmission model used for forecasting Uncertainty in the geographical spread of infections
Transitional uncertainty	Additions/withdrawals to/from vaccine inventories, with noise from refrigeration failures
Control uncertainty	Which population groups were vaccinated given the planned prioritzation How vaccines were allocated relative to the plan
Implementation uncertainty	Deviations when vaccines are not given to the correct people
Communication errors	Reporting errors from the field Failure to notify people when they should be vaccinated
Goal uncertainty	Disagreements in prioritizing who should be vaccinated first
Political/regulatory uncertainty	If/when a vaccine will be approved Allocation of vaccines to different states, countries

Table 10.1 Illustration of different types of uncertainty arising in the vaccination response to the COVID pandemic

- Historical data - A common strategy is to simply run a process over historical data. This is widely used to test trading strategies in finance, for example, where this is known as "back testing."

- Observational sampling - This is where we use observations from an exogenous process, most commonly referred to as the "real world," to generate sample realizations.

- Numerical simulations - We may have a (typically large) computer model of a complex process. The simulation may be of a physical system such as a supply chain or an asset allocation model. Some simulation models can require extensive calculations (a single sample realization could take hours or days on a computer). We can use such simulations as a source of observations similar to observations from real-world environments.

- Contingencies - We use the term "contingency" to refer to outcomes that *may* happen, and we have to plan for the possibility that they may happen, without building a probability model or estimating the frequency of these events. For example, companies managing power grids are required to plan for the event that their largest generator may fail. Some will use the term "scenario" to refer to a contingency, but "scenarios" are often used to refer to samples of a set of random variables, which are used to represent a sample of a probability distribution.

Often, we create simulated versions of the real world in order to test algorithms, with the understanding that the simulated source of observations will be replaced with exogenous observations. It is important to understand whether this is the eventual plan, since some policies depend on having access to an underlying model.

10.3.2 Types of distributions

While it is easy to represent random information as a single variable such as W_t, it is important to realize that random variables can exhibit very different behaviors. The major classes of distributions that we have encountered in our work include:

- Exponential (or geometric) families of random variables, which are the most familiar - These include the continuous distributions such as normal (or Gaussian) distributions, log normal, exponential and gamma distributions, and discrete distributions such as the Poisson distribution, geometric distribution, and the negative binomial distributions. We also include in this class the uniform distribution (continuous or discrete).

- Heavy-tailed distributions - Price processes are a good example of variability that tends to exhibit very high standard deviations. An extreme example is the Cauchy distribution which has infinite variance, but there can be less extreme distributions with heavy tails.

- Spikes - These are infrequent but extreme observations. For example, electricity prices periodically spike from typical prices in the range of 20 to 50 dollars per megawatt, to prices of 300 to as much as 10,000 dollars per megawatt for very short intervals (perhaps 5 to 10 minutes).

- Bursts - Bursts describe processes such as snow or rain, power outages due to extreme weather, or sales of a product where a new product, advertising or price reduction can produce a rise in sales over a period of time. Bursts are characterized by a sequence of observations over a short period of time.

- Rare events - Rare events are similar to spikes, but are characterized not by extreme values but rather by events that may happen, but happen rarely. For example, failures of jet engines are quite rare, but they happen, requiring that the manufacturer hold spares.

- Regime shifting - A data series may move from one regime to another as the world changes. For example, the discovery of fracking created a new supply of natural gas which resulted in electricity prices dropping from around $50 per megawatt-hour to around $20 per megawatt-hour.

- Hybrid/compound distributions - There are problems where a random variable is drawn from a distribution with a mean which is itself a random variable. The mean of a Poisson distribution, perhaps representing people clicking on an ad, might have a mean which itself is a random variable reflecting the behavior of competing ads.

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$
ω^n	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
ω^1	45.00	45.53	47.07	47.56	47.80	48.43	46.93	46.57
ω^2	45.00	43.15	42.51	40.51	41.50	41.00	39.16	41.11
ω^3	45.00	45.16	45.37	44.30	45.35	47.23	47.35	46.30
ω^4	45.00	45.67	46.18	46.22	45.69	44.24	43.77	43.57
ω^5	45.00	46.32	46.14	46.53	44.84	45.17	44.92	46.09
ω^6	45.00	44.70	43.05	43.77	42.61	44.32	44.16	45.29
ω^7	45.00	43.67	43.14	44.78	43.12	42.36	41.60	40.83
ω^8	45.00	44.98	44.53	45.42	46.43	47.67	47.68	49.03
ω^9	45.00	44.57	45.99	47.38	45.51	46.27	46.02	45.09
ω^{10}	45.00	45.01	46.73	46.08	47.40	49.14	49.03	48.74

Table 10.2 Illustration of a set of sample paths for prices all starting at \$45.00.

10.3.3 Modeling sample paths

In chapter 9, section 9.8.2, we showed that we could write the value of a policy as

$$F^\pi = \mathbb{E}\sum_{t=0}^{T} C(S_t, X_t^\pi(S_t)). \tag{10.8}$$

We then wrote this as a simulation using

$$F^\pi(\omega) = \sum_{t=0}^{T} C(S_t(\omega), X_t^\pi(S_t(\omega))), \tag{10.9}$$

where the states are generated according to $S_{t+1}(\omega) = S^M(S_t(\omega), X_t^\pi(S_t(\omega)), W_{t+1}(\omega))$. In this section, we illustrate our notation for representing sample paths more carefully.

We start by assuming that we have constructed 10 potential realizations of price paths $p_t,\ t = 1, 2, \ldots, 8$, which we have shown in table 10.2. Each sample path is a particular set of outcomes of the p_t for all time periods. We index each potential set of outcomes by ω, and let Ω be the set of all sample paths where, for our example, $\Omega = \{1, 2, \ldots, 10\}$. Thus, $p_t(\omega^n)$ would be the price for sample path ω^n at time t. For example, referring to the table we see that $p_2(\omega^4) = 45.67$.

One reason that we may generate information on the fly is that it is easier to implement in software. For example, it avoids generating and storing an entire sample path of observations. However, another reason is that random information may depend on the current state, a setting we address next.

10.3.4 State/action dependent processes

Imagine that we are looking to optimize an energy system in the presence of increasing contributions from wind and solar energy. It is reasonable to assume that the available energy from wind or solar, which we represent generically as W_t, is not affected by any decision we make. We could create a series of sample paths of wind, which we could denote by $\hat{\omega} \in \hat{\Omega}$, where each sequence $\hat{\omega}$ is a set of outcomes of $W_1(\hat{\omega}), \ldots, W_T(\hat{\omega})$. These sample paths could be stored in a dataset and used over and over.

There are a number of examples where exogenous information depends on the state of the system. Some examples include:

EXAMPLE 10.18

A drone is monitoring a forest for evidence of fires. What the drone observes (the exogenous information) depends on its location (its state).

EXAMPLE 10.19

Imagine the setting where a patient is being given a cholesterol lowering drug. We have to decide the dosage (10mg, 20mg, . . .), and then we observe blood pressure and whether the patient experiences any heart irregularities. The observations represent the random information, but these observations are influenced by the prior dosage decisions.

EXAMPLE 10.20

The price of oil reflects oil inventories. As inventories rise, the market recognizes the present of surplus inventories which depresses prices. Decisions about how much oil to store affects the exogenous changes in market prices.

In some cases, the random information depends on the decision being made at time t. For example, imagine that we are a large investment bank buying and selling stock. Large buy and sell orders will influence the price. Imagine that we place a (large) order to sell x_t shares of stock, which will clear the market at a random price

$$p_{t+1}(x_t) = p_t - \theta x_t + \varepsilon_{t+1},$$

where θ captures the impact of the order on the market price. We are not able to directly observe this effect, so we create a single random variable $\hat{p}_{t+1}$ that captures the entire change in price, given by

$$\hat{p}_{t+1} = -\theta x_t + \varepsilon_{t+1}.$$

Thus, our random variable $\hat{p}_{t+1}$ depends on the decision x_t.

We can model problems where the exogenous information W_{t+1} depends on the action x_t as if it were depending on the post-decision state $S_t^x = (S_t, x_t)$. However, since it is the sales x_t itself that influences the change in price, it is important that x_t be captured explicitly in the post-decision state.

Whether the exogenous information depends on the state or the action, it depends on the policy, since the state at time t reflects prior decisions.

10.3.5 Modeling correlations

One of the most difficult problems in stochastic modeling is capturing correlations. Some examples of types of correlations include:

- Correlations over time - Activities from one time period to the next can be positively correlated (increased demand suggests that the demand in the next time period may be even higher) or negatively correlated (above average observations will be followed by below average observations).

- Correlation over space - There are many problems that exhibit strong spatial correlations. Some examples include:
  - Weather - Temperature, wind speed, and rainfall will tend to show strong positive correlations with distance.
  - Presence of disease - Since diseases spread from one person (or animal) to another, the result is spatial pockets of disease that tend to grow.
  - Purchasing behavior - Word of mouth about a product may produce spatial pockets of similar buying behavior.
- Correlation based on characteristics or features - We might see similarities in how people respond to a type of medication based on gender, genetic markets, or smoking history. We might be modeling market demands for similar products.

One of the challenges when generating random samples when there are correlations is that we may have to capture these correlations at different levels of aggregation. We note that the hierarchical aggregation methodology presented in section 3.6.1 accomplishes this automatically.

10.4 MONTE CARLO SIMULATION

We now address the problem of generating random variables from known probability distributions using a process known as Monte Carlo simulation. Although most software tools come with functions to generate observations from major distributions, it is often necessary to customize tools to handle more general distributions.

There is an entire field that focuses on developing and using tools based on the idea of Monte Carlo simulation, and our discussion should be viewed as little more than a brief introduction.

10.4.1 Generating uniform $[0,1]$ random variables

Arguably the most powerful tool in the Monte Carlo toolbox is the ability to use the computer to generate random numbers that are uniformly distributed between 0 and 1. This is so important that most computer languages and computing environments have a built-in tool for generating uniform $[0,1]$ random variables, as well as random variables from other distributions. While we strongly recommend using these tools, it is useful to understand how they work. It starts with a simple recursion that looks like

$$R^{n+1} \leftarrow (a + bR^n) \bmod (m),$$

where a and b are very large numbers, while m might be a number such as $2^{64}-1$ (for a 64 bit computer), or perhaps $m = 999,999,999$. For example, we might use

$$R^{n+1} \leftarrow (593845395 + 2817593R^n) \quad \bmod (999999999).$$

This process simulates randomness because the arithmetic operation $(a + bR)$ creates a number much larger than m, which means we are taking the low order digits, which move in a very random way.

We have to initialize this with some starting variable R^0 called the *random number seed*. If we fix R^0 to some number (say, 123456), then every sequence $R^1, R^2, \ldots$ will be exactly the same (some computers use an internal clock to keep this from happening, but sometimes this is a desirable feature). If a and b are chosen carefully, R^n and R^{n+1} will appear (even under careful statistical testing) to be independent.

Due to the $\mod$ function, all the values of R^n will be between 0 and 999999999. This is convenient because it means if we divide each of them by 999999999, we get a sequence of numbers between 0 and 1. Thus, let

$$U^n = \frac{R^n}{m}.$$

While this process looks easy, we caution readers to use built-in functions for generating random variables, because they will have been carefully designed to produce the required independence properties. Every programming language comes with this function built in. For example, in Excel, the function `Rand()` will generate a random number between 0 and 1 which is both uniformly distributed over this interval, as well as being independent (a critical feature).

Below, we are going to exploit our ability to generate a sequence of uniform $[0, 1]$ random variables to generate a variety of random variables which we denote $W^1, \ldots, W^n, \ldots$. We refer to the sequence W^n as a Monte Carlo sample, while modeling using this sample is referred to as Monte Carlo simulation.

There is a wide range of probability distributions that we may draw on to simulate different types of random phenomena, so we are not even going to attempt to provide a comprehensive list of probability distributions. However, we are going to give a summary of some major classes of distributions, primarily as a way to illustrate different methods for generating random observations.

10.4.2 Uniform and normal random variable

Now that we can generate random numbers between 0 and 1, we can quickly generate random numbers that are uniform between a and b using

$$X = a + (b - a)U.$$

Below we are going to show how we can use our ability to generate (0,1) random variables to generate random variables from many other distributions. However, one important exception is that we cannot easily use this capability to generate random variables that are normally distributed.

For this reason, programming languages also come with the ability to generate random variables Z that are normally distributed with mean 0 and variance 1. With this capability, we can generate random variables that are normally distributed with mean μ and variance σ^2 using the sample transformation

$$X = \mu + \sigma Z.$$

We can take one more step. While we will derive tremendous value from our ability to generate a sequence of *independent* random variables that are uniformly distributed on $[0, 1]$, we often have a need to generate a sequence of *correlated* random variables that are normally distributed. Imagine that we need a vector X

$$X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{pmatrix}.$$

Now assume that we are given a covariance matrix Σ where $\Sigma_{ij} = Cov(X_i, X_j)$. Just as we use σ above (the square root of the variance σ^2), we are going to take the "square root" of Σ by taking its Cholesky decomposition, which produces an upper right-triangular matrix. In Python (using the `numpy` package), this can be done using

$$C = \texttt{numpy.linalg.cholesky}(\Sigma).$$

The matrix C satisfies

$$\Sigma = CC^T,$$

which is why it is sometimes viewed as the square root of Σ.

Now assume that we generate a column vector Z of N independent, normally distributed random variables with mean 0 and variance 1. Let μ be a column vector of $\mu_1, \ldots, \mu_N$ which are the means of our vector of random variables. We can generate a vector of N random variables X with mean μ and covariance matrix Σ using

$$\begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_N \end{pmatrix} + C \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_N \end{pmatrix}.$$

To illustrate, assume our vector of means is given by

$$\mu = \begin{bmatrix} 10 \\ 3 \\ 7 \end{bmatrix}.$$

Assume our covariance matrix is given by

$$\Sigma = \begin{bmatrix} 9 & 3.31 & 0.1648 \\ 3.31 & 9 & 3.3109 \\ 0.1648 & 3.3109 & 9 \end{bmatrix}.$$

The Cholesky decomposition computed in Python using $C = \texttt{numpy.linalg.cholesky}(\Sigma)$ is

$$C = \begin{bmatrix} 3 & 1.1033 & 0.0549 \\ 0 & 3 & 1.1651 \\ 0 & 0 & 3 \end{bmatrix}.$$

Imagine that we generate a vector Z of independent standard normal deviates

$$Z = \begin{bmatrix} 1.1 \\ -0.57 \\ 0.98 \end{bmatrix}.$$

Using this set of sample realizations of Z, a sample realization u would be

$$u = \begin{bmatrix} 10.7249 \\ 2.4318 \\ 9.9400 \end{bmatrix}.$$

10.4.3 Generating random variables from inverse cumulative distributions

Assume we have a distribution with density $f_X(x)$ and cumulative distribution $F_X(x)$, and let $F_X^{-1}(u)$ be the inverse, which means that $x = F_X^{-1}(u)$ is the value of x such that the probability that $X \leq x$ is equal to u (it helps if $0 \leq u \leq 1$). There are some distributions where $F_X^{-1}(u)$ can be found analytically, but computing this numerically can also be quite practical. We now use the following trick from probability. Let U be a random variable that is uniform over the interval $[0, 1]$. Then $X = F_X^{-1}(U)$ is a random variable that has the distribution $X \sim f_X(x)$.

A simple example of this result is the case of an exponential density function $\lambda e^{-\lambda x}$ with cumulative distribution function $1 - e^{-\lambda x}$. Setting $U = 1 - e^{-\lambda x}$ and solving for x gives

$$X = -\frac{1}{\lambda}\ln(1 - U).$$

Since $1 - U$ is also uniformly distributed between 0 and 1, we can use

$$X = -\frac{1}{\lambda}\ln(U).$$

We can generate outputs from a gamma distribution given by

$$f(x|k,\theta) = \frac{x^{k-1}e^{-\frac{x}{\theta}}}{\theta^k\Gamma(k)}.$$

$\Gamma(k)$ is the gamma function, with $\Gamma(k) = (k-1)!$ if k is integer. The gamma distribution is created by summing k exponential distributions, each with mean $(k\lambda)^{-1}$. This can be simulated by simply generating k random variables with an exponential distribution and adding them together.

A special case of this result allows us to generate binomial random variables. First sample U which is uniform on [0,1], and compute

$$R = \begin{cases} 1 & \text{if } U < p \\ 0 & \text{otherwise.} \end{cases}$$

R will have a binomial distribution with probability p. The same idea can be used to generate a geometric distribution, which is given by (for $x = 0, 1, \ldots$)

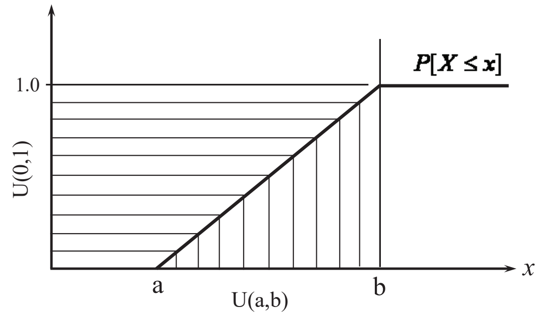
$$\mathbb{P}(X \leq x) = 1 - (1-p)^{k+1}.$$

Now generate U and find the largest k such that $1 - (1-p)^{k+1} \leq U$.

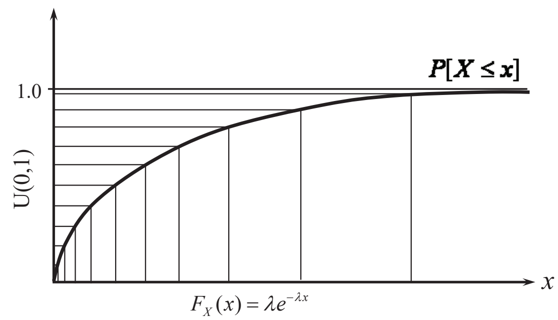
Figure 10.1 illustrates using the inverse cumulative-distribution method to generate both uniformly distributed and exponentially distributed random numbers. After generating a uniformly distributed random number in the interval [0,1] (denoted $U(0,1)$ in the figure), we then map this number from the vertical axis to the horizontal axis. If we want to find a random number that is uniformly distributed between a and b, the cumulative distribution simply stretches (or compresses) the uniform (0,1) distribution over the range (a, b).

10.4.4 Inverse cumulative from quantile distributions

This same idea can be used with a quantile distribution (which is a form of nonparametric distribution). Imagine that we compile our cumulative distribution from data. For example,



10.1a: Generating uniform random variables.



10.1b: Generating exponentially-distributed random variables.

Figure 10.1 Generating uniformly and exponentially distributed random variables using the inverse cumulative distribution method.

we might be interested in a distribution of wind speeds. Imagine that we collect a large sample of observations $X_1, \ldots, X_n, \ldots, X_N$, and further assume that they are sorted so that $X_n \leq X_{n+1}$. We would then let $F_X(x)$ be the percentage of observations that are less than or equal to x. The inverse cumulative is computed by simply associating $f_n = F_X(x_n)$ with each observation x_n. Now, if we choose a uniform random number U, we simply find the smallest value of n such that $f_n \leq U$, and then output X_n as our generated random variable.

10.4.5 Distributions with uncertain parameters

Imagine that we have the problem of optimizing the price charged for an airline or hotel given the random requests from the market. It is reasonable to assume that the arrival process is described by a Poisson arrival process with rate λ customers per day. However, in most settings we do not know λ.

One approach is to assume that λ is described by yet another probability distribution. For example, we might assume that λ follows a gamma-distribution, which is parameterized by (k, θ). Now, instead of having to know λ, we just need to choose (k, θ), which are referred to as *hyperparameters*. Introducing a belief on unknown parameters introduces more parameters for fitting a distribution. For example, if λ is the expected number of arrivals per day, then the variance of the number of arrivals is also λ, but it is quite likely that the variance is much higher. We can tune the hyperparameters (k, θ) so that we still match the mean but produce a variance closer to what we actually observe.

Consider, for example, the problem of sampling Poisson arrivals describing the process of booking rooms for a hotel for a particular date. For simplicity, we are going to assume that the booking rate is a constant λ over the interval $[0, T]$ where T is the date where people would actually stay in the room (in reality, this rate would vary over time). If N_t is the number of customers booking rooms on day t, the probability distribution of N_t would be given by

$$\mathbb{P}[N_t = i] = \frac{\lambda^i e^{-\lambda}}{i!}.$$

We can generate random samples from this distribution using the methods presented earlier.

Now assume that we are uncertain about λ. We might assume that it has a beta distribution which is given by

$$f(x : \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1},$$

where $\Gamma(k) = (k-1)!$ (if k is integer). The beta distribution takes on a variety of shapes over the domain $0 \leq x \leq 1$ (check out the shapes on Wikipedia). Assume that when we observe bookings, we find that N_t has a mean μ and variance σ^2. If the arrival rate λ were known, we would have $\mu = \sigma^2 = \lambda$. However, in practice we often find that $\sigma^2 > \mu$, in which case we can view λ as a random variable.

To find the mean and variance of λ, we start by observing that

$$\mathbb{E}N_t = \mathbb{E}\{\mathbb{E}\{N_t|\lambda\}\} = \mathbb{E}\lambda = \mu.$$

Finding the variance of λ is a bit harder. We start with the identity

$$\begin{aligned} Var N_t &= \sigma^2 \\ &= \mathbb{E}N_t^2 - (\mathbb{E}N_t)^2. \end{aligned} \tag{10.10}$$

This allows us to write

$$\begin{aligned} \mathbb{E}N_t^2 &= Var N_t + (\mathbb{E}N_t)^2 \\ &= \sigma^2 + \mu^2. \end{aligned}$$

We then use

$$\begin{aligned} \mathbb{E}N_t &= \mathbb{E}\{\mathbb{E}\{N_t|\lambda\}\} \\ &= \mathbb{E}\lambda, \\ &= \mu. \\ \mathbb{E}N_t^2 &= \mathbb{E}\{\mathbb{E}\{N_t^2|\lambda\}\} \\ &= \mathbb{E}\{\lambda + \lambda^2\} \\ &= \mu + (Var\lambda + \mu^2). \end{aligned}$$

We can now write

$$\begin{aligned} \sigma^2 + \mu^2 &= \mu + (Var\lambda + \mu^2), \\ Var\lambda &= \sigma^2 - \mu. \end{aligned}$$

So, given the mean μ and variance σ^2 of N_t, we can find the mean and variance of λ.

The next challenge is to find the parameters α and β of our beta distribution, which has mean and variance

$$\begin{aligned}\mathbb{E}X &= \frac{\alpha}{\alpha+\beta},\\ \mathit{Var}X &= \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}.\end{aligned}$$

We are going to leave as an exercise to the reader to decide how to pick α and β so that the moments of our beta-distributed random variable X match the moments of λ.

The parameters α and β are called *hyperparameters* as they are distributional parameters that describe the uncertainty in the arrival rate parameter λ. α and β should be chosen so that the mean of the beta distribution closely matches the observed mean μ (which would be the mean of λ). Less critical is matching the variance, but it is important to reasonably replicate the variance σ^2 of N_t.

Once we have fit the beta distribution, we can run simulations by first simulating a value of λ from the beta distribution. Then, given our sampled value of λ (call it $\hat{\lambda}$), we would sample from our Poisson distribution using arrival rate $\hat{\lambda}$.

10.5 CASE STUDY: MODELING ELECTRICITY PRICES

With the emphasis on renewables, there has been considerable interest in modeling the stochastic processes that arise in this setting. In this section, we will look at challenges that arise when modeling the price of electricity purchased from the grid, and the energy from a wind farm.

We begin with the problem of modeling real-time electricity prices, shown in figure 10.2. These prices, taken from the grid operated by PJM Interconnections, which operates the grid serving the mid-Atlantic states in the United States. The prices are from February, 2015, and illustrate the well-known heavy-tailed behavior of electricity prices.

The most elementary model for prices is a basic random walk, given by

$$p_{t+1} = p_t + \varepsilon_{t+1}, \tag{10.11}$$

where we typically assume that $\varepsilon_{t+1} \sim N(0, \sigma_\varepsilon^2)$, which is estimated from the sequence of observations of $p_{t+1} - p_t$.

There are a number of problems with this model for applications such as electricity prices. The remainder of this section will suggest methods to improve the performance of this basic model.

10.5.1 Mean reversion

The most popular stochastic model for prices is known most simply as a mean-reversion model, or, if you enjoy using jargon, the *Ornstein-Uhlenbeck process*. We start by tracking the mean of the process using a simple exponential smoothing model

$$\bar{\mu}_t = (1-\eta)\bar{\mu}_{t-1} + \eta p_t,$$

where η is a stepsize (or smoothing factor, or learning rate) that smooths the price signal, which is typically a number in the range $[.01, 0.10]$.

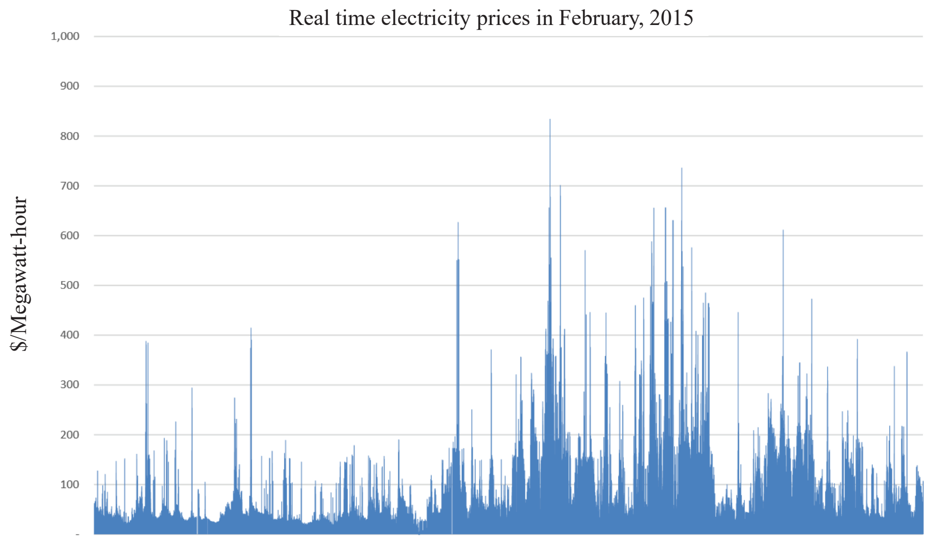


Figure 10.2 Electricity spot prices at 5-minute intervals in February, 2015 for PJM Interconnections.

Given this estimate of the mean, the mean-reversion model is given by

$$p_{t+1} = p_t + \kappa(\bar{\mu}_t - p_t) + \varepsilon_{t+1}, \tag{10.12}$$

where κ is another smoothing coefficient that has to be calibrated to produce the best fit of estimated and actual prices. If p_t is greater than the estimate of the mean $\bar{\mu}_t$, the next price is pushed down. The noise term ε_{t+1} is typically assumed to be normally distributed with distribution $N(0, \sigma_\varepsilon^2)$, where σ_ε^2 is calculated from the differences between the estimated price $\bar{p}_t$ given by

$$\bar{p}_t = p_t + \kappa(\bar{\mu}_t - p_t),$$

and the actual price p_{t+1}.

10.5.2 Jump-diffusion models

A limitation of a basic mean-reversion model is that the distribution of p_{t+1} may not be well-described by a normal distribution (given p_t). A simple fix is to use a "jump-diffusion" model, which uses two noise terms that we will call ε^{base} and ε^{jump}. We only add the jump term for a small percentage of the time periods, given by ρ^{jump}. We accomplish this by introducing the indicator variable

$$I_t^{jump} = \begin{cases} 1 & \text{with probability } \rho^{jump}, \\ 0 & \text{otherwise.} \end{cases}$$

We can now write our jump diffusion model as

$$p_{t+1} = p_t + \kappa(\bar{\mu}_t - p_t) + \varepsilon_{t+1}^{base} + I_{t+1}^{jump}\varepsilon_{t+1}^{jump}. \tag{10.13}$$

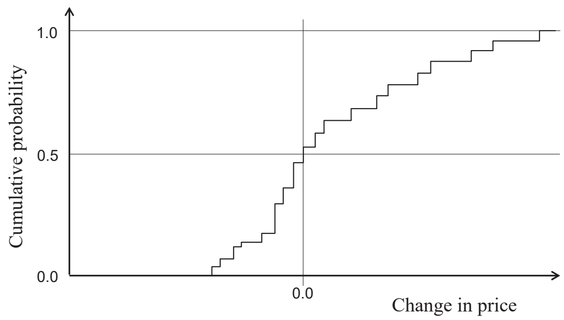


Figure 10.3 Illustration of a quantile distribution for changes in prices.

We estimate ρ^{jump} by starting with the basic mean-reversion model in equation (10.12), fitting the model, then estimating the variance σ_ε^2. Then, we pick a tolerance (say three standard deviations), and classify any price p_{t+1} that differs from the predicted price $\bar{p}_t$ by more than three standard deviations, and classifying this price as falling outside of the base model. The fraction of prices falling in this range then gives us an initial estimate of ρ^{jump}.

We then estimate the distribution of the jump noise ε_{t+1}^{jump} by just using those points that fall outside of the three-sigma range, and also re-estimate the distribution of ε_{t+1}^{base} using only the prices that fall within the three-sigma range. Of course, the variance of the error distribution for this subset will be smaller than before. As a result, we would normally repeat the process using the jump diffusion model (10.13). This process might be repeated several times until the estimates stabilize.

The jump diffusion model produces an error

$$\varepsilon_{t+1} = \varepsilon_{t+1}^{base} + I_{t+1}^{jump}\varepsilon_{t+1}^{jump}$$

that will better approximate heavy-tailed behavior than a simple normal distribution.

10.5.3 Quantile distributions

A common problem in many applications (such as electricity prices) is asymmetric distributions. The largest prices are much larger relative to the mean than the smallest prices. The same is also true with energy from wind, since gusts of wind can be much larger relative to the mean than zero, which is the smallest wind speed. In addition, choosing a parametric distribution that fits either of these processes is challenging.

An alternative approach to using parametric distributions such as the normal is to compile the cumulative distribution of errors directly from the data. A quantile distribution is shown in figure 10.3, which illustrates its ability to capture asymmetric, heavy-tailed behavior. This is a form of nonparametric distribution (which is also a lookup table), since you have to store $F_X(x)$ for each possible value of x. So, if prices range from 0 to \$1,000, and we want to store the cumulative distribution in increments of 0.10, we need a table with possibly 10,000 different values (although we only have to store the cumulative distribution for prices we actually observe).

It is relatively easy to store the cumulative distribution in a more compact way. The bigger problem is when the distribution depends on other variables such as temperature and humidity. If we divide temperature into 10 ranges, and humidity into 10 ranges, then we have 100 combinations of temperature and humidity, and we would need to compute a cumulative distribution for each of these 100 combinations (this is a classic curse-of-dimensionality since we are using a lookup table for temperature and humidity). Parametric distributions may offer more compact strategies for incorporating additional dependent variables, but this is generally not possible when using lookup tables (that is, the quantile distributions).

All the methods above focus on getting a better representation of the distribution of errors, but ignore the correlations across time. High electricity prices tend to come in bursts, correlated with hot days where temperatures stay high for periods of time. We propose two approaches, regime shifting and crossing times, that address this issue.

10.5.4 Regime shifting

A powerful strategy is to identify "regimes" that describe different ranges of our random variable (such as price). For example, we may divide prices into five ranges, or regimes. Each regime might be associated with combinations of temperature and humidity (or any other exogenous variable), or it can be ranges of prices.

First, we compute the distributions we are interested in (such as the change of price) indexed by regime. So, rather than enumerating 100 combinations of temperature and humidity, we could group these into five or 10 buckets that we think best explain prices. Number the regimes $s_1, \ldots, s_K$ and let $\mathcal{S}^{regime}$ be the set of regimes. Then, we have two tasks:

- Compute error distributions (and any other quantities) using any methodology indexed by the regime you are in. These distributions may be parametric or nonparametric (e.g. quantiles), using any of the modeling strategies described above.
- Add up the number of times f_{s_k,s_ℓ} that you transition from regime s_k to regime s_ℓ, and then normalize these to obtain the transition probabilities

$$P^{regime}_{s_k,s_\ell} = Prob[S^{regime}_{t+1} = s_\ell | S^{regime}_t = s_k]. \tag{10.14}$$

Both of these sets of calculations are performed while stepping forward in time through historical data. If your regimes depend on other variables (such as humidity and temperature), then you will need this historical data as well.

Regime shifting is a form of indexed modeling, which can be thought of as a nonparametric modeling strategy. It depends on being able to identify a reasonably small number of regimes, and simplifies modeling by allowing us to fit models that work for individual regimes, rather than globally over the entire dataset.

Regime shifting also gives us another critical feature. Our jump diffusion model, for example, assumed that the jump indicator variable I^{jump}_t was independent across time periods. However, looking at the price plot in figure 10.2, we see that there are bursts of higher prices. We can capture these bursts, to a degree, since $P^{regime}_{s_k,s_k}$ will be the probability that we stay in regime s_k, allowing us to capture a certain level of persistence.

10.5.5 Crossing times

An important characteristic when modeling stochastic processes is not just capturing whether the forecast is above or below the actual, but how long it *stays* above or be-

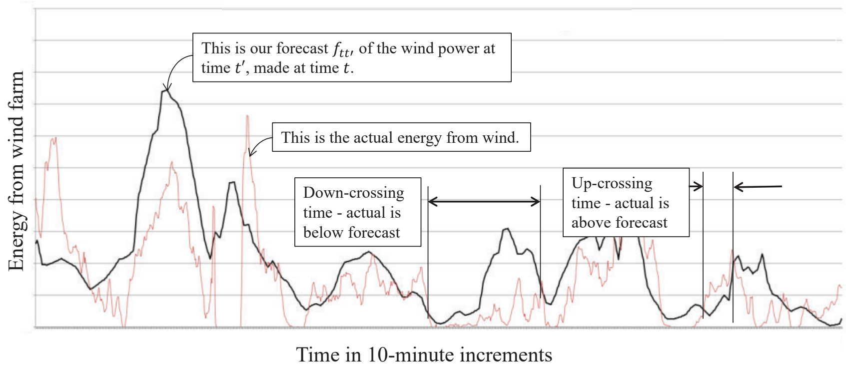


Figure 10.4 Actual vs. predicted energy from wind, showing up- and down-crossings.

low. This is important in many settings. For example, if the price of electricity stays high for a period of time, then a utility that has to pay this price may run out of cash reserves.

We are going to use the context of simulating energy from wind. Figure 10.4 shows a sample path of actual energy from wind, along side the forecast (made, say, at noon the day before). The figure shows two time intervals: the first where the actual is below the forecast, called a "down-crossing time," and the second where the forecast is above the forecast, called the "up-crossing time." These "crossing times" are periods where the actual is continuously below or above the forecast.

We are going to build on the methods we have described above in this section to develop a more sophisticated method for replicating how long stochastic processes stay at higher or lower levels, although this time we are going to use energy from a wind farm, where we are trying to model the errors relative to a forecast of the wind energy.

Our modeling strategy uses the following ideas:

a) As we step forward in time through the historical dataset, each time the actual crosses from above (below) the forecast to below (above) the forecast, we are going to compute the time that the actual was above or below the forecast, and classify that into a set of ranges (say three, for short (S), medium (M) or long (L)), that we will treat like regimes which consist of whether the actual was above or below (call this A or B), and then which of the time ranges that the length of the interval falls in (S, M, or L). If there are three time ranges, then there are six regimes, giving us $\mathcal{S}^{regime} = \{A-S, A-M, A-L, B-S, B-M, B-L\}$.

b) Each time we determine we have come to the end of a crossing time, we update a) our frequency counter f_{s_k,s_ℓ} for $s_k, s_\ell \in \mathcal{S}^{regime}$ and b) the frequency distribution for how long the interval lasted given the regime it was in. After normalization we obtain the regime transition matrix $P^{regime}_{s_k,s_\ell}$ and the distribution of the length of each crossing time given the regime S^{regime}.

c) For each time period where we know S^{regime}_t, aggregate the energy from wind, E_t, into a set of ranges (say five), and call this set $\mathcal{E}$.

d) Given the aggregated energy from wind at time t, $E^g_t \in \mathcal{E}$, and the crossing state $i \in \mathcal{I}$, observe the energy E_{t+1} (which is not aggregated) and compile a cumulative distribution of E_{t+1} given the crossing state i and aggregated wind state E_t^g. This distribution will look like the distribution in figure 10.3. So, if we have six crossing states and five aggregated wind speeds, this gives us 30 states, which means we are creating 30 wind speed cumulative distributions. The result of this calculation is the distribution

$$P_w^W(e,s) = Prob[W_{t+1}|E_t^g = e, S_t^{regime} = s].$$

This logic is quite powerful. We now have the ability to explicitly model the distribution of *how long* the actual data stream (wind speed in this case) stays above or below a baseline (the wind speed forecast). In other applications, the baseline could be just an average.

10.6 SAMPLING VS. SAMPLED MODELS

Monte Carlo sampling is without question the most powerful tool in our toolbox for dealing with uncertainty. In this section, we illustrate three ways of performing Monte Carlo sampling: 1) iterative sampling, 2) solving a static, sampled model, and 3) sequentially solving a sampled model with adaptive learning.

10.6.1 Iterative sampling: A stochastic gradient algorithm

Imagine that we are interested in solving the problem

$$\begin{aligned} F(x) &= \mathbb{E}F(x,W) && (10.15)\\ &= \mathbb{E}\{p\min\{x,\hat{D}\} - cx\}, && (10.16)\end{aligned}$$

where $W = \hat{D}(\omega)$ is a sample realization of the demand $\hat{D}$, drawn from a full set of outcomes Ω. We could search for the best x using a classical stochastic gradient algorithm such as

$$x^{n+1} = x^n - \alpha_n \nabla_x F(x^n, \hat{D}(\omega^{n+1})), \tag{10.17}$$

where

$$\nabla_x F(x,\hat{D}) = \begin{cases} p-c & x > \hat{D}, \\ -c & x \le \hat{D}. \end{cases} \tag{10.18}$$

$\nabla_x F(x,\hat{D})$ is called a *stochastic gradient* because it depends on the random variable $\hat{D}$. Under some conditions (for example, the stepsize α_n needs to go to zero, but not too quickly), we can prove that this algorithm will asymptotically converge to the optimal solution.

10.6.2 Static sampling: Solving a sampled model

A sampled version of this problem, on the other hand, involves picking a sample $\hat{\Omega} = \{\omega^1,\ldots,\omega^N\}$. We then solve

$$\bar{\theta}^N = \arg\min_\theta \frac{1}{N}\sum_{n=1}^N F(\theta|\omega^n). \tag{10.19}$$

This is actually a deterministic problem (known in some communities as the *sample average approximation*), although one that is much larger than the original stochastic problem (see section 4.3 for a more complete discussion). For many applications, equation (10.19) can be solved using a deterministic solver, although the problem may be quite large. The stochastic gradient update (10.17) can be much easier to compute than solving the sampled problem (10.19).

The quality of the solution to (10.19) compared to the optimal solution of the original problem (10.15) depends on the application, but as we saw in section 4.3.2, the rate of convergence of $\bar{\theta}^N$ to the optimal θ (for an infinite sample) is actually quite fast.

In practice, stochastic gradient algorithms require tuning the stepsize sequence α_n which can be quite frustrating. On the other hand, stochastic gradient algorithms can be implemented in an online fashion (e.g. through field observations) while the objective (10.19) is a strictly offline approach. There is a rich theory showing that the optimal solution of (10.19), x^N, asymptotically approaches the true optimal (that is, the solution of the original problem (10.15)) as N goes to infinity, but the algorithm is always applied to a static sample $\hat{\Omega}$. Unlike our stochastic gradient algorithm in the previous section, there is no notion of asymptotic convergence (although in practice we will typically stop our stochastic gradient algorithm after a fixed number of iterations).

10.6.3 Sampled representation with Bayesian updating

We close our discussion with an illustration of using a sampled model where we are uncertain about the parameters of the model. We then run experiments sequentially and update our belief about the probability that each sampled parameter value is correct.

Imagine, for example, that we we are solving a stochastic revenue management problem for airlines where we assume that the customers arrive according to a Poisson process with rate λ. The problem is that we are not sure of the arrival rate λ. We assume that the true arrival rate is one of a set of values $\lambda_t^1, \ldots, \lambda_t^K$, where each is true with probability q_t^k. The vector q_t captures our belief about the true parameters, and can be updated using a simple application of Bayes theorem.

Now let $N(\lambda)$ be a Poisson random variable with mean λ, and let N_{t+1} be the observed number of arrivals between t and $t+1$. We can update q_t using

$$q_{t+1}^k = \frac{\mathbb{P}(N(\lambda) = N_{t+1}|\lambda = \lambda^k) q_t^k}{\sum_{\ell=1}^K q_t^\ell \mathbb{P}(N(\lambda) = N_{t+1}|\lambda = \lambda^\ell)},$$

where

$$\mathbb{P}(N(\lambda) = N_{t+1}|\lambda = \lambda^\ell) = \frac{(\lambda^\ell)^{N_{t+1}} e^{\lambda^\ell}}{N_{t+1}!}.$$

The idea of using a sampled set of parameters is quite powerful, and extends to higher dimensional distributions. However, identifying an appropriate sample of parameters becomes harder as the number of parameters increases.

10.7 CLOSING NOTES

We could have dedicated this entire book to methods for modeling stochastic systems without any reference to decisions or optimization. The study of stochastic systems can be found under names including Monte Carlo simulation and uncertainty quantification, with significant contributions from communities that include statistics, stochastic search, simulation optimization and stochastic programming. This chapter is designed only to provide an indication of some of the topics that a reader will encounter when developing a sequential decision model.

There are a wide range of problems in energy, supply chain management, engineering and health where the process of designing a stochastic model of different sources of uncertainty is quite likely going to be harder than designing an effective policy (although this is not to minimize the importance of effective policies). As we now transition to chapter 11 on designing policies, we encourage the reader to think of the developing a stochastic model and an associated policy as an iterative process. The four classes of policies are of increasing complexity, and you may want to get a simpler policy working for the purpose of testing your software while you are building a more sophisticated uncertainty model.

10.8 BIBLIOGRAPHIC NOTES

Section 10.1 - Our identification of the different sources of uncertainty from the perspective of a model is new.

Section 10.3 - Stochastic modeling is a rich and mature field of study with a long history. For example, there is a field called *uncertainty quantification*; see Smith (2014) and Sullivan (2015) for modern introductions. Stochastic modeling is a term that is often associated with Monte Carlo simulation (see the next section).

Section 10.4 - Monte Carlo simulation is a field with a deep and rich history, starting with the basic idea of using a computer to generate seemingly random numbers. The field has matured to address all the dimensions of modeling stochastic systems. Some examples of excellent introductions are Nelson (2013), Carsey & Harden (2014), Law (2007), and Thomopoulos (2013). For a rigorous treatment of the mathematics of simulation can be found in Asmussen & Glynn (2007). There are a number of books describing these methods in the context of specific fields. For example, Glasserman (2004) and McLeish (2005) describe simulation methods for finance, while Carsey & Harden (2014) presents the methods in the context of the social sciences.

EXERCISES

Review questions

10.1 Section 10.5 describes a series of models: mean reversion, jump diffusion, quantile distributions, regime shifting, and crossing times. Very briefly summarize the *specific feature* that each of these strategies contributes relative to the most basic random walk model

$$p_{t+1} = p_t + \varepsilon_{t+1}$$

where $\varepsilon_{t+1} \sim N(0, \sigma_\varepsilon^2)$.

10.2 Section 10.5.5 models "crossing times" for a stochastic process.

a) Describe what is meant by a "crossing time."

b) The methodology is described as a form of regime shifting. What is the set of regimes introduced for the problem of modeling wind energy?

Modeling questions

10.3 For each of the forms of uncertainty below, list the category (or categories) from section 10.1 that best describe the form of uncertainty:

a) The response of a patient to a new drug.

b) The energy that will be generated by a wind farm over the next hour, E_{t+1}, given the observation of wind over each of the previous six hours, E_t, E_{t-1}, E_{t-5}, and the fitted linear model:

$$E_{t+1} = \theta_0 E_t + \ldots + \theta_5 E_{t-5} + \varepsilon_{t+1}.$$

c) The number of people who say they will vote for a candidate running for office in a telephone poll of 100 people.

d) The estimated location of a ship calculated using a radar signal, which might incur distortions from weather.

e) The performance of a dispatcher for a trucking company assigning drivers to loads.

f) The tariffs to be paid for parts imported from another country next year.

g) The number of units of inventory transferred from one store to another as instructed by a central manager.

h) The performance of each member in a team managing a portfolio of physical assets.

i) The change in market price when a large mutual fund decides to sell a large number of shares in a stock (enough to affect the market).

Computational exercises

Exercises 10.4 to 10.10 all use the electricity price data that can be downloaded from the supplementary materials website `http://tinyurl.com/RLSOsupplementary`, "Spreadsheet of electricity price data" (under Chapter 10). Use the tab for the February price data.

10.4 Electricity prices tend to be very random, with very large spikes. Start by assuming that electricity prices p_t (where t steps forward in 5-minute increments) are coming from an exponential distribution, which means we can write

$$p_t \sim \lambda e^{-\lambda y}.$$

Assume that p_t is independent of p_{t+1}. There are 288 5-minute time periods in a day.

a) Use the computed average price $\bar{p}$ (given in the spreadsheet) to compute $\lambda = 1/\bar{p}$. Then, use the cumulative distribution to compute the expected number of prices (out of the 8064 time periods in February) should be above 100, 200, ..., 500.

Compare this to the actual number of prices above each of these values (use the yellow highlighted cell to enter these values to get both the expected number of prices that are over these values, and the actual number). What pattern do you see?

b) Show how to perform a sample realization from an exponential distribution using the ability of a computer to generate a random variable U that is uniformly distributed between 0 and 1.

c) Simulate 8064 observations of prices, and plot them as we have plotted the actual prices. How do the two graphs compare?

10.5 Using the spreadsheet for electricity prices, fit a random walk model (equation (10.11)), where you will have to estimate the variance of ε_{t+1} from the 8064 prices. Generate a sample of 8064 prices using this model, and compare to the actual historical prices. How would you characterize the similarities, and differences, between the two sets of prices?

10.6 Again using the spreadsheet for electricity prices, fit a mean reversion model, where you will have to tune κ (do this using trial and error) to find them model that fits the best. Use $\eta = 0.10$ in your smoothing model for $\bar{\mu}_t$. You will also need to use the model to estimate the variance of ε_{t+1}. Finally, generate another sample of 8064 prices and compare the results to the actual prices.

10.7 Follow the instructions in section 10.5.2 to fit a jump diffusion model, and compare the results to the historical data.

10.8 Use the basic random walk model in equation (10.11) to compute the errors, and then fit a quantile distribution using price increments of \$1. Again, simulate the 8064 prices from this model, and compare the patterns with the historical model, as well as the prices from the random walk model (and other methods that you may have implemented above).

10.9 Divide the range of prices into five ranges of your choosing (these may be of equal size, but you may wish to experiment with different sizes, given the wide range of prices). Compute the regime shifting probability distribution $P^{regime}_{s_k,s_\ell}$ defined in equation (10.14). Now fit a normal distribution for the change in prices for each region. Finally, simulate the evolution of regimes, and then draw a random price for the random distribution in each regime. Compare your results to the historical prices.

10.10 Use the steps described in section 10.5.5 to estimate the regime transition probabilities and the conditional wind distributions $Prob[W_{t+1}|E^g_t = e, S^{regime}_t = s]$. Finally, use these distributions to simulate electricity prices, and compare the resulting sample (over the 8064 time periods) to history.

Theory questions

10.11 Let X be a random variable (any random variable with finite variance) and let $F_X(x)$ be the cumulative distribution, which means $F_X(x) = Prob[X \leq x]$. Let $F^{-1}(u)$, where $0 \leq u \leq 1$, be the inverse cumulative distribution, where $u = Prob[X \leq F^{-1}(u)]$. Show that the random variable U where $U = F^{-1}(X)$ is uniformly distributed between 0 and 1.

Sequential decision analytics and modeling

These exercises are drawn from the online book *Sequential Decision Analytics and Modeling* available at `http://tinyurl.com/sdaexamplesprint`.

10.12 Read chapter 8, sections 8.1 - 8.4, but our focus will be on the uncertainty modeling in section 8.3, which describes three ways of modeling uncertainty in the forecast. Describe each method in detail, and discuss the strengths and weaknesses of each method.

10.13 Read chapter 9, sections 9.1 - 9.4, but our focus will be on the uncertainty modeling in section 9.3, which describes two ways of modeling uncertainty in the forecast. Describe each method in detail, and discuss the strengths and weaknesses of each method.

Diary problem

The diary problem is a single problem you chose (see chapter 1 for guidelines). Answer the following for your diary problem.

10.14 Create your own version of table 10.1 by listing the different categories of uncertainty, and then list the types of uncertainty in your diary problem (if any) that belong to each category. You may feel that a type of uncertainty in your problem can be listed in more than one category.

Bibliography

Asmussen, S. & Glynn, P. W. (2007), *Stochastic simulation: algorithms and analysis*, Springer Science & Business Media.

Carsey, T. M. & Harden, J. J. (2014), *Monte Carlo Simulation and Resampling Methods for Social Science*, Sage Publications.

Glasserman, P. (2004), *Monte Carlo Methods in Financial Engineering*, Springer-Verlag, New York.

Law, A. M. (2007), *Simulation Modeling and Analysis*, McGraw-Hill, New York.

McLeish, D. L. (2005), *Monte Carlo Simulation and Finance*, John Wiley & Sons, New York.

Nelson, B. L. (2013), *Foundations and Methods of Stochastic Simulation: A first course*, Springer, New York.

Smith, R. C. (2014), *Uncertainty Quantification: Theory, Implementation, and Applications*, SIAM, Philadelphia.

Sullivan, T. (2015), *Introduction to Uncertainty Quantification*, Springer, New York.

Thomopoulos, N. (2013), *Essentials of Monte Carlo Simulation: Statistical methods for building simulation models*, Springer, New York.