Chapter 9

ESTIMATION AND FORECASTING

9.1 Introduction

A revenue management system requires forecasts of quantities such as demand, price sensitivity, and cancellation probabilities, and its performance depends critically on the quality of these forecasts. Indeed, some industry estimates suggest that a 20% reduction of forecast error can translate into a 1% incremental increase in revenue generated from the RM system (Poelt [424]). While it is difficult to generalize from such figures, there is little doubt that good forecasting is vitally important for RM. In practice, forecasting is a high-profile task of RM, consuming the vast majority of development, maintenance, and implementation time.

The term *forecast* may conjure up the notion of a single number, such as the demand for a specific day on a specific flight in the future or demand for a particular item at a retail store (a so-called *point estimate*). *A*. certain amount of misunderstanding about RM forecasting is not uncommon among nontechnical analysts and managers, who are accustomed to thinking of forecasts as a single number. However, a point estimate is almost never accurate; a forecast is more complicated than a single number and needs to be understood in statistical terms that account for the inherent uncertainty in predicting future outcomes.

In this chapter, we examine forecasting for RM. We start with an overview of the role of forecasting in RM—surveying the available data sources, forecasting strategies and methodologies, and factors involved in actually operationalizing a RM forecasting system. The remainder of the chapter describes estimation and forecasting methods in more depth.

While our discussion is centered on RM forecasting, most of the techniques we describe are not particular to RM and as such can safely be described as *standard*. Indeed, there are many excellent textbooks devoted to estimation and forecasting, and it is not our intention to approach them in scope and depth. Rather, this chapter is intended as a primer on the subject—sufficient in coverage to give a good sense of the range of methods and issues involved in RM forecasting but not providing an in-depth reference on any one method. We do, however, give enough detail to understand and implement at least a basic version of each method. To implement and maintain a high-quality, production-level RM forecasting system, one needs to know more about the nuances of each forecasting method, and the reader in this situation is encouraged to consult a specialized source for such information. (The Notes and Sources section at the end of the chapter contains references to a number of books dedicated to estimation and forecasting.)

9.1.1 The Forecasting Module of RM Systems

RM forecasting presents many challenges to a system designer. For one, a significant amount of programming work is involved in collecting and manipulating the data to convert it into the required data feeds for the forecasting module. Large volumes of transactional data have to be gathered from multiple sources, either in real time or on an overnight batch basis. The database design is an important issue because in many large-scale implementations, an immense number of records have to be retrieved, updated, and added within a small time window. Data back-up procedures take up further time. All these data and systems issues must be addressed prior to actual forecasting itself.

Figure 9.1 shows a schematic of the process flow of a typical quantity-based RM system and where the forecasting module resides in the process. The outputs of the forecasting module are fed to the optimization module, which produces RM controls such as markdown prices, booking limits, bid prices, and overbooking limits. In the stage between forecasting and optimization, most RM systems also give analysts monitoring and overriding ability over the forecasts. These so-called *user influences* are used to either increase or decrease the forecasts at different levels of aggregation before they are used in optimization. Indeed, in most quantity-based RM systems, analysts are not permitted to change capacity controls directly but can change them only indirectly by manipulating the forecast inputs. This practice is based on the belief (widespread among RM practitioners) that knowledge of markets or special conditions can sometimes make human analysts better than algorithms at forecasting demand, but rarely, if ever, can human analysts set RM controls better than optimization algorithms.

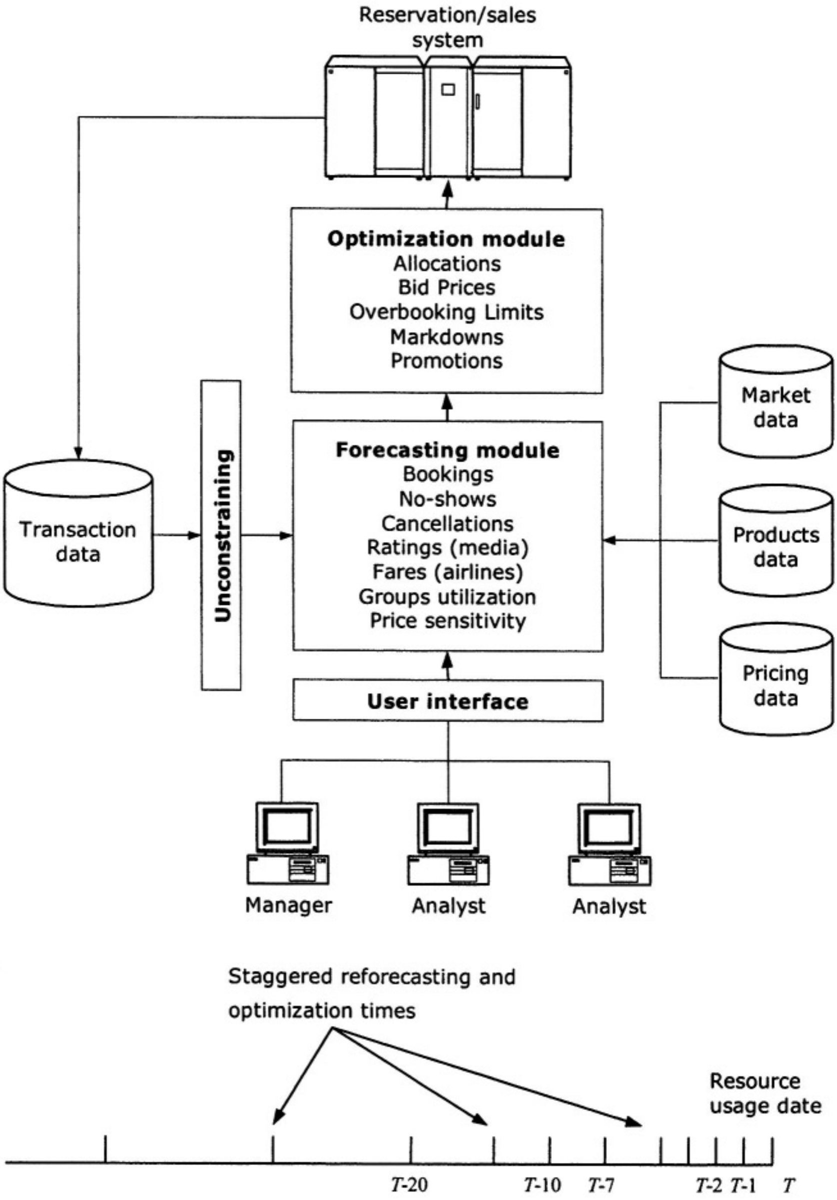


Figure 9.1. Forecasting module in a RM system. Periodic reforecasting and optimization timeline.

In most RM systems, forecasting is automated, transactional, and data-driven—as opposed to qualitative (such as expert opinion) or survey-based. This is primarily due to the sheer volume of forecasts that have to be made and the tight processing-time requirements. These practical constraints limit the choice of forecasting algorithms. They also limit the types of data that can reasonably be collected and the amount of time a user can spend calibrating and verifying forecasts. Often, certain forecasting procedures, even if they give superior forecasts, may not be a viable option because they take too long to run, require data that is too expensive to collect (say, using surveys), or require too much expert, manual effort to calibrate.

For quantity-based RM, most systems use time-series methods, which use historical data to project the future. For price-based RM systems (retail RM, for example), one is usually interested in forecasting demand as a function of marketing variables such as price or promotion. As a result, causal forecast models, which use explanatory variables such as prices, weather or economic indicators, play a bigger role in price-based RM.

9.1.2 What Forecasts Are Required?

RM forecasting requirements are driven by the input requirements of the optimization module. As we saw in previous chapters, most optimization models use stochastic models of demand and hence require an estimate of the complete probability distribution or at least parameter estimates (e.g., means and variances) for an assumed distribution. Moreover, forecasting aggregate demand is just one of a host of quantities that need to be estimated in a RM system. Many other features of the demand—how it evolves over time, what percentage cancel, how it responds to a promotion—are also important in making good control decisions, and need forecasting.

Quantity-Based RM Forecasts Quantity-based RM industries like airlines and hotels have a wide variety of forecasting requirements. For example, in addition to the demand data, characterizing the way reservations for different customer types arrive during the booking period is important for some optimization models. Thus, so-called *booking-curve* or *booking-profile forecasting* is usually an important task.

Cancellation and no-show probabilities usually have to be estimated as well. Cancellation probabilities tend to be a function of time. (A customer who books early may have a higher probability of cancelling than one who books later.) Therefore, forecasting a cancellation *curve* over time may be more appropriate, giving better information to the optimization module. No-shows occur at the time of service; hence, assuming that a customer shows up with a certain probability is often an appropriate model of no-shows (see Chapter 4). Both no-show and cancellation-rate forecasts have to be calculated for future customers as well as customers who have already booked. For existing reservations, however, additional sources of information, such as the customer's own past history of cancelling, whether the reservation has been paid and ticketed, and characteristics of how the reservation was made (channel, time, etc.) can be used, increasing the data-gathering requirements.

Revenue values are also critical inputs to optimization modules. Often, these values change over time or are uncertain, so the prices at which the products will be sold in the future may have to be forecasted. Predicting revenue values can be a major challenge, especially when prices change rapidly and competitive forces drive pricing.

Optimization models may also require estimates of cross-selling and up-selling probabilities. The buy-up factors discussed in Section 2.6.1 may have to be estimated from historical data. Spill and recapture are two other quantities that are sometimes required in setting (or at least managing) RM controls. *Spill* refers to the amount of demand that is lost by closing down a class or because a compartment is sold out, while *recapture* is the amount of this spilled demand that is recaptured by the firm's substitute products. The discrete-choice model of Section 2.6.2, requires estimates of the parameters of a choice model, sometimes by channel of distribution or by customer segment.

Price-Based RM Forecasts For price-based RM, somewhat different forecasts are required. One common requirement is an estimate of the parameters of a demand function—or at least an estimate of the price sensitivity at the current price. Cross-price elasticity estimates may also be required when there are significant substitution effects (say, for category pricing in a retail store), which vastly increases the scale of the forecasting task. In addition, forecasts of demand at the current price, the size of the potential customer population, stockout and low-inventory effects, and switching behavior may be required. Such estimates require looking at the historical price-demand relationship of the product or similar products or at intertemporal panel tracking data. Some retailers have also tried intelligent experimentation in real time to estimate how consumers will respond to various price changes (*live price testing*).

In summary, the forecasting requirements in even a modestly large RM system are daunting, indeed. It is little wonder, then, that developing a good forecasting system is so vitally important for a successful RM implementation.

9.1.3 Data Sources

Data is the life-blood of a forecasting system. Therefore, identifying which sources of data are available and how they can best be used is an important first step in developing a forecasting system.

Most RM systems in practice rely primarily on historical sales data to construct forecasts. While this leads to highly efficient systems for data collection, forecast calibration, and automated forecasting, relying on historical data has its weaknesses. For example, in industries where products change frequently—when an airline offers service to a new city or at a new time for instance—there is often little historical data on which to base forecasts. Similarly, in media RM, forecasts for rating of new programs must be constructed despite the fact that their ratings often have little relationship to the ratings observed for past programs. Fashion apparel retailers, for instance, have to estimate sales of new styles that may be only vaguely similar to the styles sold in the past. In addition, even if the product stays constant, major changes in the economy, competing technologies, or industry structure may render past data of little use in predicting the future.

In short, if no explicit relationships with external data sources are tracked, the forecasting system will be "blind" to outside events. The same is true with respect to changes in competitors' products and prices. In cases where such external data is ignored, it is common practice in RM to rely on analysts to monitor outside events and compensate by adjusting forecasts appropriately through so-called *user influences.*

9.1.3.1 Sales-Transaction Data Sources

The main sources of data in most RM systems are transactional databases—for example, reservation and property management systems (PMSs), CRM and ERP databases, and retail inventory and scanner databases. Further descriptions of these data sources are given in Chapters 10 and 11. These sources may be centralized, independent entities shared by other firms in the industry (such as GDSs of the airline industry selling MIDT data), a centralized facility within a company that interfaces with several local systems (a retail chain's point-of-sale (POS) system linking all its stores), a local reservation system (a hotel PMS), or customer-oriented databases with information on individual customers and their purchase history (customer-relationship management (CRM) systems and PNR databases).

For quantity-based RM, the most widely used transactional data source is the reservations database. Reservation databases typically store customer data in two formats: either as an aggregate number of bookings in a class (total bookings) or as information about each individual booking, called a *customer booking record* (*passenger name record (PNR)* in the airline industry). Forecasts may be based on either the aggregate bookings or individual customer booking records. The aggregate bookings data contain information only on the total number booked in each fare class, while the individual booking records contain much more specific information on each customer—such as their name, address, booking time, number of units booked, amount paid, frequent flyer or other loyalty program number, booking class, cancellation time, capacity used (length of stay and room number for a hotel, car type and duration for a rental-car company, or itinerary for an airline), ancillary spending (dining expenses, telephone calls). The customer record may also contain links to other customer records (for example, for a group booking) that may be useful for forecasting.

For retail RM, factory-shipment data, store-level scanner data, consumer-panel data, regional demographic data, and advertising and promotions data are the primary data used. Industry-wide aggregate scanner data (sold by firms such as Information Resources, Inc. and A.C. Nielsen). Warehouse-shipment data can be obtained from Selling Areas Marketing, Inc. (SAMI), which provides sales, average price, and distribution information for the U.S.

Panel data, obtained from tracking purchases of a group of panelists over time, provides valuable information on cross-sectional and intertemporal purchase behavior. Such data are widely used in retail and media industries. A panel member's purchase data is also linked to promotions, availability, displays, advertising, couponing, and markdowns through the time of purchase, allowing for precise inferences on preferences and marketing influences. Many marketing research companies provide such panel-data services.

9.1.3.2 Controls-Data Sources

In addition to sales information, databases often store information on the controlling process itself. Examples of this kind of data include records of when a class is closed for further bookings, snapshots of bid prices used in the control, overbooking authorizations, past prices, and promotion activities. Such information is of great use in correcting for unobservable no-purchase decisions by potential customers (Section 9.4).

Industrywide database systems (such as a GDSs in the travel industry) may also yield additional information for forecasting—for example, the availability of competitor bookings, prices of competing products, and market share. Many airline GDS companies make this information available on a weekly or monthly basis on tapes called *market infor-*

mation data tapes (MIDT). Although few airlines at present use this information in their RM systems, it is useful for estimating competitive market share, and for longer-term, strategic planning and analysis.

For markdown pricing and other price-based RM applications, the control decisions are past history of prices and promotions. Most retail POS systems store this information routinely. For in-store displays or bundle pricing, the POS data has to be merged with a marketing database. Inventory data is also provided by many retail POS systems, and this data is useful for correcting for stockouts and broken-assortments effects (missing color-size combinations).

9.1.3.3 Auxiliary Data Sources

A few auxiliary data sources also play a big role in RM forecasting in some industries. For instance, currency exchange-rate and tax information is necessary to keep track of revenue value for sales in different countries. In the airline industry, the schedules and possible connections (provided by firms such as OAG) are required to determine what markets are being served. If a ticket is sold across multiple airlines, the various prorating agreements affect the ultimate revenue value of each product sold. A revenue accounting system keeps track of such agreements and calculates the net revenue.

In broadcasting, ratings, customer location, and demographic information is required. A causal forecasting method may take into account information on the state of the economy, employment, income and savings rates, among other factors. A rental-car firm can use advance travel bookings to predict its own demand at airport locations. Information on ad-hoc events (special events) like conferences, sports events, concerts, holidays, is also crucial in improving the accuracy of forecasts. Many forecasting systems allow the users to manually enter information on such events.

Many retail RM systems also use weather data, which is supplied by several independent vendors via daily automated feeds. Short-term weather forecasts guide discounting and stocking decisions (for example, a snow-storm could suggest high demand for snow shovels). Weather data also plays an important role in energy forecasting for electric power generators and distributors.

Macroeconomic data (such as GNP growth rates and housing starts) is rarely used in automated, tactical forecasting but frequently plays a role in aggregate forecasts of factors such as competitors' costs, industry demand and market share, and broad consumer preferences. Statistics on cost of labor are published by the Bureau of Labor Statistics (BLS) in the United States in a monthly publication called *Employment and*

Earnings, which provides average hourly earnings for workers by product category. BLS also provides monthly producer price indexes on raw materials.

For products sold through distributors, important data is not always available. For example, an automobile manufacturer may not know the final price paid by a customer because dealers have no obligation to report this information back to the manufacturer. Similarly, trade promotions may lead to increased shipments for the manufacturer, but the distributor may simply stock up during the trade deal and sell at a normal price, reducing the impact of the promotion. Lack of such information is one of the impediments for many firms contemplating RM.

9.1.3.4 Partial-Bookings Data

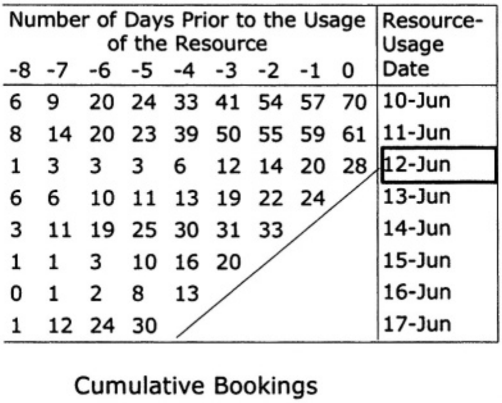
In most quantity-based RM applications, bookings occur over an extended period of time, yet the product or service is provided on a very frequent basis, often daily. For example, an airline may sell seats on a flight that operates every day, but bookings can occur over a period of 12 months prior to departure; hotels take reservations for rooms for each day, yet bookings are made many days or weeks in advance. In such situations, there are often large quantities of so-called *partial-booking data* in the reservation system. While incomplete, such data is quite useful for forecasting.

Figure 9.2 shows an example of partial-bookings data, indicating the number of bookings observed each day for capacity in the past as well as the future. The y**-axis** represents the date of service (such as the departure date in the airline case or the check-in date for a hotel), and the x**-axis** represents the number of days prior to the date of the service.

One way to use these partial histories of bookings is to forecast the *increments* of demand for each booking day, rather than forecasting the total demand to come. Thus, for example, data on the demand received on the 12^{th} day prior to service can be used to predict demand on the 12^{th} day prior to service in the future, even though the data may be from a booking history that is incomplete. Such methods are discussed in more detail in Section 9.3.9.

9.1.4 Design Decisions

After the data sources are identified, one has to make a number of design decisions regarding the forecasting strategy and methodology. Here we look at the main design decisions in qualitative terms.



Number of Days Prior to the Usage of the Resource									Resource-Usage
-8	-7	-6	-5	-4	-3	-2	-1	0	Date
6	9	20	24	33	41	54	57	70	10-Jun
8	14	20	23	39	50	55	59	61	11-Jun
1	3	3	3	6	12	14	20	28	12-Jun
6	6	10	11	13	19	22	24		13-Jun
3	11	19	25	30	31	33			14-Jun
1	1	3	10	16	20				15-Jun
0	1	2	8	13					16-Jun
1	12	24	30						17-Jun

Cumulative Bookings

Figure 9.2. Wedge shape of cumulative current-bookings data for a sequence of usage dates on each of the days prior to usage.

9.1.4.1 Parametric or Nonparametric Forecasts?

As mentioned in Section 9.1.2, in most RM forecasting we are interested in estimating a probability distribution of future demand—or in estimating demand as a function of price variables or product attributes. Such estimates can be made in one of two ways. The first is to assume a specific functional form and then estimate the parameters of this functional form. This approach is called *parametric estimation.* Alternatively, distributions or functions can be estimated directly based on observed historical data, without assuming any a *priori* functional form. This approach is called *nonparametric estimation.* Choosing between a parametric or nonparametric approach to forecasting is a basic design decision.

While nonparametric methods are in a sense more general, they are not necessarily a better choice. Nonparametric estimates suffer from two serious drawbacks: First, because they do not use a functional form to "fill in" for missing values, they often require much more information than is available in many RM applications to obtain reasonable estimates of a distribution or demand function. Second, even with sufficient data, nonparametric estimates may not be as good at predicting the future, even if they fit the historical data well. Parametric models are better able to "smooth out" the noise inherent in raw data, which often results in a more robust forecast. Indeed, we know of no RM systems that use purely nonparametric methods to estimate demand, though several use nonparametric methods in selected places. Neural networks, which are sometimes viewed as *semiparametric* methods, have been reported in several RM applications, and these we cover later in this chapter.

Parametric methods usually are much more modest in their data requirements, have the advantage of providing estimates of demand that extend beyond the range of the observed data (allow for extrapolation), and are generally more robust to errors and noise in the data. The disadvantage of parametric techniques is that some properties of the distribution must be assumed—for example, that it is symmetric about the mean, has certain coefficients of variation, or has certain *tail behavior* (the characteristics of the demand distribution for extreme values of demand). Thus, parametric methods can suffer in terms of overall forecasting accuracy if the actual demand distribution deviates significantly from these assumptions (called *specification errors*). Because they are more widely used in RM, we focus on parametric methods in this chapter.

9.1.4.2 Levels of Aggregation

Forecasts can be made at different levels of aggregation as well, and how to aggregate data and forecasts is another important design decision. To give an example, airlines price their products by fare-basis codes with a large number of fares-basis codes sold within each booking class. Capacity control, however, is usually performed at the booking-class level. How, then, should forecasting be handled? Should the demand be forecast for each fare product (basis code) or each booking class? That is, should we aggregate all the fare products in a booking class and forecast at the level of the booking class? Or should we forecast at the fare-product level and aggregate these forecasts into a forecast for booking-class demand?

Another level-of-aggregation design decision comes up in network RM (Chapter 3), where the optimization system requires a forecast of demand for each multiresource product in the network. In principle, the forecasts should be at the level of the network products (O&Ds or lengths of stay) as this is the level required by network-optimization models. However, many reservation systems do not collect data at this level of detail. In the airline case, for example, reliable data may exist only for individual flight legs, and we may have to heuristically disaggregate leg-level forecasts into O&D, product-level forecasts.

Ultimately, however, we need to produce the forecasts required by the optimization module. Continuing the airline example, this would imply generating forecasts at the fare-product level if we were using a bid-price control, but perhaps at the booking-class level if we were using a resource level, booking-class-based control. On the other hand, the requirements of the optimization module can often be manipulated. For example, one can simply forecast at the booking-class level and then assume that all the demand in this booking class has the same (say, the average) fare. In the network case, the RM system might be using a simple single-resource heuristic to approximate the network RM problem, in which case an aggregate forecast for each resource independently may do just fine. Thus, the forecasting and optimization design decisions are intimately related. Indeed, in practice it is hard to change one without affecting the other.

In retail RM as well, the level of aggregation in forecasting is largely governed by the data and optimization requirements. Store-level pricing requires store-level estimates of demand and price sensitivity for each product, whereas a model that optimizes prices set on a chainwide basis may not require this same level of detail. If household purchase data (panel data) is available or if experiments or surveys can be conducted, then one can forecast based on models of individual purchase behavior and combine these to determine an aggregate demand function. However, if only aggregate POS sales data can be obtained, one has to estimate the aggregate demand function directly.

9.1.4.3 Bottom-Up versus Top-Down Strategies

Broadly speaking, there are two different strategies for aggregating forecasting. In a *bottom-up forecasting strategy,* forecasting is performed at a detailed level to generate *subforecasts.* The end forecast is then constructed by aggregating these detailed subforecasts. In a *top-down forecasting strategy,* forecasts are made at a high level of aggregation—a *superforecast*—and then the end forecast is constructed by disaggregating these superforecasts down to the level of detail required. The following are examples of bottom-up and top-down forecasting strategies.

Example 9.1 (BOTTOM-UP FORECASTING) An airline is interested in getting forecasts of load factors (occupancy) for each flight in each compartment for an upcoming season. The airline stores data of each past customer, itinerary, and fare class. This itinerary-level data is first used to make forecasts for the number of customers expected to book for each itinerary and fare-class combination for every day of the season. Next, these detailed forecasts are added together to produce an aggregate forecast for the seasonal load-factors.

Example 9.2 (TOP-DOWN FORECASTING) A hotel is interested in a forecast of the number of people expected to book for each future date in each room-category and length-of-stay combination. The hotel first forecasts the total number of guests who will book to arrive on each day in each rate category (the superforecast). Then it forecasts the fraction of guests that stay for a specified length of time (a length-of-stay distribution). Finally, it combines these two components to arrive at an end forecast of expected number who will start their stay on a specific date and stay a certain number of days by multiplying the forecast for the aggregate number of guests on a specific date by the estimate of the fraction that will stay for a given length of time.

Which strategy is most appropriate is not always clear-cut. It depends on the data that is available and accessible to an automated system on a daily basis, the outputs required, and the types of forecasts already being made and available for use. Moreover, the "right" answer in most cases is that *both* strategies are required because certain phenomena can be estimated only at a low level of aggregation, while others can be estimated only at a high level of aggregation.

For example, it is clear that in an airline network if one wants an estimate of demand for each itinerary and fare-class combination, then aggregate booking-class or flight-leg data will not be sufficient; data for each passenger itinerary is required. At the same time, such passenger-level data is sparse, with often only a handful of bookings occurring for any given combination in a year. Hence, aggregate phenomena such as daily or weekly seasonalities, holiday effects, or upward or downward trends in total demand are—for all practical purposes—unobservable at the disaggregate level; one must look at aggregate booking data over many itinerary and fare-class combinations to observe such effects. Even with good passenger-level data, one may therefore need to aggregate data and perform aggregate forecasts to identify important "large-scale" phenomena. As a result, hybrid combinations of bottom-up and top-down approaches are the norm in practice.

9.2 Estimation Methods

Estimation is the problem of finding model parameters that best describe a given set of observed data. Forecasting, in contrast, involves predicting future, unobserved values. Thus, estimation is generally *descriptive* (characterizing what *has been* observed), while forecasting is *predictive* (characterizing what *will be* observed). Roughly, in the RM context, estimation is the calibration of a forecasting model's parameters (hence it also is called *forecast calibration*) and is done relatively infrequently; while forecasting is the use of the estimated model to predict future values, and is performed frequently on an operational basis.

For example, an estimate of price sensitivity based on past sales data may be used in a forecast of future demand. Similarly, many forecasting methods are based on estimating the parameters of a dynamic model from historical demand data, which is subsequently used to predict future values of demand. Yet this distinction between estimation and forecasting is not always very sharp. In some methods, such as the Kalman filtering, estimation and forecasting work in lock-step, one after another.

Here we examine methodology for parameter estimation and discuss some of the theoretical and practical issues that arise.

9.2.1 Estimators and Their Properties

An estimator represents, in essence, a formalized “guess” about the parameters of the underlying distribution from which a sample (the observed data) is assumed to be drawn. Estimators can take on many forms and can be based on different criteria for a “best” guess. We use demand estimation as our example, but the ideas apply more generally.

9.2.1.1 Nonparametric Estimators

Let the random variable Z_k denote the k^{th} observation of demand, and let $F(z) = P(Z_k \leq z)$ denote the distribution of Z_k. Nonparametric estimation methods do not make any assumptions on the underlying distribution $F(\cdot)$. For example, we could estimate $F(z)$ by simply computing the fraction of observations in the sample that are less than or equal to z for each value of z. This empirical distribution then forms a nonparametric estimate of the true distribution $F(z)$.

Nonparametric estimates of this type have the advantage of not requiring any assumptions on the form of the distribution. However, as mentioned earlier, they typically require more data to produce accurate estimates and do not allow one to extrapolate beyond the observed data easily. For example, if there were no observations less than 10 in a data set, then the empirical distribution would estimate that $P(Z_k \leq z) = 0$ for all values of z less than 10.

9.2.1.2 Parametric Estimators

For a parametric estimator, we assume that the underlying distribution of Z_k is of the form

$$P(Z_k \leq z | \boldsymbol{\beta}, \mathbf{y}_k) = F(z, \boldsymbol{\beta}, \mathbf{y}_k), \tag{9.1}$$

where $\mathbf{y}_k = (y_{k1}, \ldots, y_{kM})$ is a vector of M explanatory (independent) variables (time, indicators of holiday events, prices, lagged observations of Z_k itself, and so on) and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_M)$ is a M-dimensional vector of parameters. For ease of exposition, we assume that the dimension of $\boldsymbol{\beta}$ and $\mathbf{y}_k$ are the same, though this is not necessary.

Assume we have a sequence of N independent observations $z_1, \ldots, z_N$, with values y_{km}, $k = 1, \ldots, N$, $m = 1, \ldots, M$ for the explanatory variables, alternatively represented by vectors $\mathbf{y}_1, \ldots, \mathbf{y}_N$ or by a $N \times M$ matrix $\mathbf{Y} = [y_{km}]_{k=1,\ldots,N,\ m=1,\ldots,M}$. The estimation problem, then, is to determine the unknown parameters $\boldsymbol{\beta}$ using only the sample of the N observations $\{Z_k\}$ (the data) and the values of the explanatory variables $\mathbf{Y}$ corresponding to each observation (characteristics of the observed data).

It is usually convenient to express the relationship between the demand and the explanatory variables by a simple functional form consisting of a systematic (deterministic) component and an additive noise component:[1]

$$Z_k = \zeta(\boldsymbol{\beta}, \mathbf{y}_k) + \xi_k, \tag{9.2}$$

where the randomness comes from the error term ξ_k. Many of the regression and time-series forecasting models of this chapter can be viewed as manifestations of (9.2). The following is a common example of (9.2):

Example 9.3 (LINEAR MODEL) Consider the linear model of demand

$$Z_k \quad = \quad \boldsymbol{\beta}^{\mathsf{T}}\mathbf{y}_k + \xi_k, \quad k = 1, \ldots, N, \tag{9.3}$$

where ξ_k are i.i.d. $N(0, \sigma^2)$ random variables, independent also of the explanatory variables $\mathbf{y}$. Z is often referred to as the *dependent* variable and the vector $\mathbf{y}$ as the *independent* variables. The distribution of Z in terms of (9.1) is then

$$F_{Z_k}(z|\boldsymbol{\beta}, \mathbf{y}_k) = P(Z_k \leq z|\boldsymbol{\beta}, \mathbf{y}_k) = \Phi(\frac{z - \boldsymbol{\beta}^{\mathsf{T}}\mathbf{y}_k}{\sigma}),$$

where $\Phi(\cdot)$ is the standard normal distribution.

9.2.1.3 Properties of Estimators

If the N observations, $\mathbf{z}_N = (z_1, \ldots, z_N)$, are considered independent realizations of $\mathbf{Z}_N = (Z_1, \ldots, Z_N)$, then the estimator based on these observations is a function of N i.i.d. random variables, $\hat{\boldsymbol{\beta}}(\mathbf{Z}_N)$, and is therefore itself a random variable. What properties would we like this (random) estimator to have?

Bias For one, it would be desirable if the expected value of the estimator equaled the actual value of the parameters—that is, if

$$E[\hat{\boldsymbol{\beta}}(\mathbf{Z}_N)] = \boldsymbol{\beta}.$$

If this property holds, the estimator is said to be an *unbiased estimator,* otherwise, it is a *biased estimator.* The estimator of the m^{th} parameter, $\hat{\beta}_m$, is said to have a *positive bias* if its expected value exceeds β_m, and a *negative bias* if its expected value is less than β_m.

If the estimator is unbiased only for large samples of data—that is, it satisfies

$$\lim_{N \to \infty} E[\hat{\boldsymbol{\beta}}(\mathbf{Z}_N)] = \boldsymbol{\beta},$$

then it is called an *asymptotically unbiased estimator.* All unbiased estimators are, of course, also asymptotically unbiased.

[1] We drop the notation conditioning on $\boldsymbol{\beta}$ and y, $(\cdot|\boldsymbol{\beta}, \mathbf{y})$, when it is obvious from the context.

Efficiency An estimator $\hat{\boldsymbol{\beta}}(\mathbf{Z})$ is said to an *efficient estimator* if it is unbiased and the random variable $\hat{\boldsymbol{\beta}}(\mathbf{Z})$ has the smallest variance among all unbiased estimators. Efficiency is desirable because it implies the variability of the estimator is as low as possible given the available data. The Cramer-Rao bound[2] provides a lower bound on the variance of *any* estimator, which can be used to prove an estimator is efficient. In particular, if an estimator achieves the Cramer-Rao bound, then we are guaranteed that it is efficient. An estimator can be inefficient for a finite sample but *asymptotically efficient* if it achieves the Cramer-Rao bound when the sample size is large.

Consistency An estimator is said to be *consistent* if for any $\delta > 0$,

$$\lim_{N \to \infty} P(|\hat{\boldsymbol{\beta}}(\mathbf{Z}_N) - \boldsymbol{\beta}| < \delta) = 1,$$

that is, if it converges in probability to the true value $\boldsymbol{\beta}$ as the sample size increases. Consistency assures us that with sufficiently large samples of data, the value of $\boldsymbol{\beta}$ can be estimated arbitrarily accurately.

Ideally, we would like our estimators to be unbiased, efficient, and consistent, but this is not always possible. We revisit these properties in Section 9.5.1.2 on specification errors.

9.2.2 Minimum Square Error (MSE) and Regression Estimators

One class of estimators is based on the *minimum mean-square error* (MSE) criterion—also referred to as *regression estimators.* MSE estimators are most naturally suited to the case where the forecast quantity has an additive noise term as in (9.2). Given a sequence of observations $z_1, \ldots, z_N$ and associated vectors of explanatory variable values $\mathbf{y}_1, \ldots, \mathbf{y}_N$, the MSE estimate of the vector $\boldsymbol{\beta}$ is the solution to

$$\min_{\boldsymbol{\beta}} \sum_{k=1}^{N} [z_k - \zeta(\boldsymbol{\beta}, \mathbf{y}_k)]^2, \tag{9.4}$$

where $\zeta(\boldsymbol{\beta}, \mathbf{y}_k)$ is as defined in (9.2). The minimization problem (9.4) can be solved using standard nonlinear optimization methods such as conjugate-gradient or quasi-Newton. However, the problem is greatly simplified if the function $\zeta(\boldsymbol{\beta}, \mathbf{y}_k)$ and the error terms have a specialized form, as shown next.

Ordinary Least-Squares (OLS) and Linear-Regression Estimators If the function ζ in (9.1), the error terms ξ_k in (9.2), and explana-

[2] See DeGroot [151], pp. 420–430 for a discussion of the Cramer-Rao bound.

tory variables $\mathbf{y}_k$ satisfy the assumptions listed in Table 9.1, the MSE estimates are also known as the *ordinary least-squares (OLS) estimators*—or *linear-regression estimators.* Specifically, suppose the observations Z_k are linear functions of M explanatory variables of the form,

$$Z_k = \boldsymbol{\beta}^{\mathsf{T}}\mathbf{y}_k + \xi_k.$$

Furthermore, suppose the explanatory variables $\mathbf{y}_k$ are uncorrelated and the error term ξ_k are independent, normal random variables that have means of zero and identical variances (homoscedasticity). Then the OLS estimators are the values $\boldsymbol{\beta}$ that solve

$$\min_{\boldsymbol{\beta}} \sum_{k=1}^{N} \left[z_k - \boldsymbol{\beta}^{\mathsf{T}}\mathbf{y}_k\right]^2 .$$

We can write equation (9.2) in matrix form as

$$\mathbf{Z} \;=\; \mathbf{Y}\boldsymbol{\beta} + \boldsymbol{\xi}, \qquad (9.5)$$

where $\mathbf{Y} = [y_{km}]_{k=1,\ldots,N,\ m=1,\ldots,M}$, and $\mathbf{Z} = (Z_1, \ldots, Z_N)$. The MSE estimates for β, given N observations $\mathbf{z} = (z_1, \ldots, z_N)$, are then

$$\hat{\boldsymbol{\beta}} \;=\; (\mathbf{Y}^{\mathsf{T}}\mathbf{Y})^{-1}(\mathbf{Y})^{\mathsf{T}}\mathbf{z}, \qquad (9.6)$$

assuming the matrix $\mathbf{Y}^{\mathsf{T}}\mathbf{Y}$ is invertible.

Example 9.4 Consider the following model of demand:

$$Z_k \;=\; \beta + \xi_k. \qquad (9.7)$$

This model has one scalar parameter β, which is constant over time, and is equivalent to having $M = 1$ and $\mathbf{Y} = (1,\ldots,1)$ in (9.5). Assume ξ_k is normally distributed with mean 0 and constant variance. Then if we have N observations, $z_1, \ldots, z_N$, the MSE estimate $\hat{\beta}$ based on this data solves

$$\min_{\beta} \sum_{k=1}^{N} [z_k - \beta]^2.$$

Applying (9.6) and noting that $\mathbf{Y}^{\mathsf{T}}\mathbf{Y} = N$ and $\mathbf{Y}^{\mathsf{T}}\mathbf{z} = \sum_{k=1}^{N} z_k$, weobtain

$$\hat{\beta} = \frac{1}{N}\left(\sum_{k=1}^{N} z_k\right),$$

which is simply the sample mean of the data.

Table 9.1. Assumptions of ordinary least-squares (OLS) estimation.

Assumption	*Violation*	*Test*	*Fix*
$\zeta(\boldsymbol{\beta}, \mathbf{y}_k)$ Linear	Nonlinear relationship	Specification tests (Section 9.5.1.2)	Transformations, Nonlinear Regression
Homoscedasticity (ξ's have constant variance across observations)	Heteroscedasticity (Variances of ξ's different for different observations)	White [563]	Transformations; GLS or MLE estimation
Errors ξ's are uncorrelated across observations	Serial correlation of observations	Durbin-Watson [166–168]; von Neumann ratio test [243]	GLS, MLE
Errors ξ's are normally distributed	Non-normal observed errors	Shapiro-Wilk	Transformations; MLE
Explanatory variables $\mathbf{y}$ are uncorrelated	Multicollinearity (some of the elements of $\mathbf{y}$ are strongly related)	Belsley-Kuh-Welsch test [44]	Drop some of the variables

From (9.6) it can be seen that the OLS estimator $\hat{\boldsymbol{\beta}}$ is a linear function of the random observations $\mathbf{Z}$, which makes computing the estimates quite easy. In addition, the OLS estimators have several desirable properties: they are consistent, unbiased, and efficient under very general conditions. For these reasons, the MSE/linear-regression estimator is popular in practice.

Regression is widely used in price-based management for estimating price sensitivity, market shares, and the effects of various marketing variables (such as displays and promotions) on demand. Regression estimates are somewhat less common in quantity-based RM forecasting application such as airline and hotel RM because in these applications it is often difficult to obtain data on the exogenous explanatory variables as an automated data feed. When regression is used in quantity-based RM, typically the only explanatory variables in the model are the historical demand data itself (the explanatory variables are past demand observations). However, in such cases formal time-series models of the type discussed in Section 9.3.2 are usually preferred.

When any of the assumptions of the OLS regression in Table 9.1 is violated, one has to resort to more advanced regression techniques such as *generalized least squares* (GLS), *seemingly unrelated regressions* (SUR), and two-stage and three-stage least squares (2SLS, 3SLS) (see Greene [220]). A description of these methods is beyond the scope of this chapter.

9.2.3 Maximum-Likelihood (ML) Estimators

While regression is based on the least-squares criterion, *maximum-likelihood (ML) estimators* are based on finding the parameters that maximize the "likelihood" of observing the sample data, where *likelihood* is defined as the probability of the observations occurring. More precisely, given a probability-density function f_Z of the process generating Z_k, $k = 1, \ldots, N$, which is a function of a vector of parameters $\boldsymbol{\beta}$ and the observations of the explanatory variables, $\mathbf{y}_k$, the likelihood of observing value z_k as the k^{th} observation is given by $f_Z(z_k|\boldsymbol{\beta}, \mathbf{y}_k)$. The likelihood of observing the N observations $(z_1, \mathbf{y}_1), \ldots, (z_N, \mathbf{y}_N)$ is then

$$\mathcal{L} = \prod_{k=1}^{N} f_Z(z_k|\boldsymbol{\beta}, \mathbf{y}_k). \tag{9.8}$$

The ML estimation problem is to find a $\boldsymbol{\beta}$ that maximizes the likelihood $\mathcal{L}$. It is more convenient to maximize the log-likelihood, In $\mathcal{L}$, because this converts the product of functions in (9.8) to a sum of functions. Since the log function is strictly increasing, maximizing the log-likelihood is equivalent to maximizing the likelihood. This gives the ML problem:

$$\max_{\boldsymbol{\beta}} \sum_{k=1}^{N} \ln f_Z(z_k|\boldsymbol{\beta}, \mathbf{y}_k).$$

In special cases, this problem can be solved in closed form. Otherwise, if the function $f_Z(\cdot)$ is a differentiable function, gradient-based optimization methods such as Newton's method can be used to solve it numerically.

ML estimators have good statistical properties under very general conditions; they can be shown to be consistent, asymptotically normal, and asymptotically efficient, achieving the Cramer-Rao lower bound on the variance of estimators for large sample sizes.

Example 9.5 (ESTIMATING THE MEAN OF A NORMAL DISTRIBUTION) Consider the following model of demand from Example 9.4:

$$Z_k = \beta + \xi_k. \tag{9.9}$$

Recall that the model assumes that the scalar parameter β is constant over time, and ξ_k is normally distributed with mean 0 and constant variance σ. Suppose we have N observations, $z_1, \ldots, z_N$. Then the ML estimator solves

$$\max_{\beta} \prod_{k=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} e^{-(z_k-\beta)^2/2\sigma^2}.$$

Taking the log of the objective function yields

$$\max_{\beta} -\frac{N\ln(2\pi)}{2} - N\ln(\sigma) - \sum_{k=1}^{N} \frac{(z_k-\beta)^2}{2\sigma^2}.$$

Differentiating with respect to β and setting the result to zero, one can show that the ML estimator is

$$\hat{\beta} = \frac{1}{N}\sum_{k=1}^{N} z_k,$$

which is just the sample mean. Note despite the fact that the estimation criterion is different, this estimator is the same as the MSE estimator of Example 9.4.

Example 9.6 (ESTIMATING THE PARAMETERS OF MULTINOMIAL-LOGIT MODEL) The MNL discrete-choice model is described in Section 7.2.2.3. The data consists of a set of N customers and their choices made from a finite set S of alternatives. Associated with each alternative j is a vector $\mathbf{y}_j$ of explanatory variables (assume for simplicity there are no customer-specific characteristics). The probability that a customer selects alternative i is then given by (assuming that all customers face the same choice-set of products)

$$P_i(S) = \frac{e^{\boldsymbol{\beta}^{\mathsf{T}}\mathbf{y}_i}}{\sum_{j\in S} e^{\boldsymbol{\beta}^{\mathsf{T}}\mathbf{y}_j} + 1}, \tag{9.10}$$

where $\boldsymbol{\beta}$ is a vector of (unknown) parameters. Let $c(k)$ be the choice made by customer k. The likelihood function is then

$$\mathcal{L} = \prod_{k=1}^{N} \left[\frac{e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{y}_{c(k)}}}{\sum_{j \in S} e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{y}_j} + 1} \right].$$

The maximum-likelihood estimate $\hat{\boldsymbol{\beta}}$ is then determined by solving

$$\max_{\boldsymbol{\beta}} \ln \mathcal{L}. \tag{9.11}$$

While this maximum-likelihood problem cannot be solved in closed form, it has good computational properties. Namely, there are closed-form expressions for all first and second partial derivatives of the log-likelihood function, and it is jointly concave in most cases (McFadden [372]; Hausman and McFadden [244]). The ML estimator has also proved to be robust in practice. (See Ben-Akiva and Lerman [48] for further discussion and case examples.)

9.2.4 Method of Moments and Quantile Estimators

While MSE and ML estimators are the most prevalent, several other estimators are also used in practice. Two common ones are the *method of moments* and *quantile estimators.*

In the method of moments, one equates moments of the theoretical distribution to their equivalent empirical averages in the observed data. This yields a system of equations that can be solved to estimate the unknown parameters $\boldsymbol{\beta}$. The following example illustrates the idea:

Example 9.7 (Estimating the parameters of a normal distribution) Suppose we want to estimate the parameters of a normal distribution. The sample mean and sample second moment are computed as follows:

$$\bar{z} = \frac{1}{N} \sum_{k=1}^{N} z_k$$

$$\overline{z^2} = \frac{1}{N} \sum_{k=1}^{N} z_k^2.$$

Equating these to the theoretical mean and second moment yields the system of equations

$$\begin{aligned} \bar{z} &= \mu \\ \overline{z^2} &= \sigma^2 + \mu^2. \end{aligned}$$

Solving for μ and σ gives the estimates $\hat{\mu} = \bar{z}$ and $\hat{\sigma} = \sqrt{\overline{z^2} - (\bar{z})^2}$.

Alternatively, we can use quantile estimates based on the empirical distribution to estimate the parameters $\boldsymbol{\beta}$ of a distribution. For example, we might estimate the mean of a normal distribution by noting that as the normal distribution is symmetric, the mean and median are the same. Hence, we can estimate the mean by computing the median of a sequence of N observations. More generally, one can compute a number of quantiles of a data set and equate these to the theoretical quantiles of the parametric distribution. In general, if m parameters need to estimated, m different quantiles are needed to produce m equations in m unknowns (for a normal distribution, for example, one could equate the 0.25 and 0.75 quantiles of the data to the theoretical values to get two equations for the mean and variance). Quantile estimation techniques are sometimes preferred, as they tend to be less sensitive to outlier data than are MSE and ML estimators.

9.2.5 Endogeneity, Heterogeneity, and Competition

Table 9.1 lists the standard problems associated with classical regression—correlation of the error terms, collinearity, and so on—and techniques for dealing with violations of the assumptions. Such problems and their corrective measures are well known and can be found in many standard econometric books. In this section, we focus on a few nonstandard estimation problems that are of particular importance for RM applications—endogeneity, heterogeneity, and competition.

9.2.5.1 Endogeneity

The model (9.2) is said to suffer from endogeneity if the error term ξ is correlated with one of the explanatory variables in $\mathbf{y}$. This is a common problem in RM practice, both in aggregate-demand function estimation and in disaggregate, discrete-choice model estimation.

For example, products may have some unobservable or unmeasurable features—quality, style, reputation—and the selling firm typically prices its products accordingly. So if there are two firms in the market with similar products and one has higher nonquantifiable quality, we may observe that the firm with the higher-quality product has both a larger market share and a higher price. A naive estimation based on market shares that ignores the unobserved quality characteristics would lead to the odd conclusions that higher price leads to higher market share! Such effects are widespread in price-elasticity estimation because we can rarely observe all relevant product and firm characteristics and price is usually correlated with many of these unobservable characteristics.

Econometricians call this problem *endogeneity* or *simultaneity.* The technical definition is that the random-error term in (9.2) is correlated with one of the explanatory variables, $E[\mathbf{Y}^{\mathsf{T}}\boldsymbol{\xi}] \neq 0$, or equivalently (in the case of linear regression) these vectors are not orthogonal. So while $\boldsymbol{\xi}$ is supposed to represent all unobservable customer and product characteristics that influence demand for a given set of explanatory variables $(\mathrm{Z}|\mathbf{y})$, some of the explanatory variables $\mathbf{y}$ also contain information on the unobservable attributes through their correlation with $\boldsymbol{\xi}$.

Econometric techniques to correct endogeneity fall under a class of methods called *instrumental-variables (IV) techniques,* attributed to Reiersøl [438] and Geary [202]. Two-stage and three-stage least-squares methods (2SLS and 3SLS) are some of the popular IV techniques. Instrumental variables are exogenous variables that are correlated with an explanatory variable but are uncorrelated with the error term ξ. If there are such IVs, we can use them to "remove" the problematic correlation between the independent variables $\mathbf{y}$ and $\boldsymbol{\xi}$.

We illustrate the idea for the case of linear regression. In (9.5) suppose

$$E[\mathbf{Y}^{\mathsf{T}}\boldsymbol{\xi}] \neq 0.$$

However, suppose there exist M instrumental variables (we can use some of the y's to construct this vector of IVs) for each observation so that we have a $N \times M$ matrix $\mathbf{V}$ with the property that $E[\mathbf{V}^{\mathsf{T}}\boldsymbol{\xi}] = 0$, and $E[\mathbf{V}^{\mathsf{T}}\mathbf{Y}]$ is nonsingular. Then the IV estimator is

$$\hat{\boldsymbol{\beta}}_{\mathrm{IV}} = [\mathbf{V}^{\mathsf{T}}\mathbf{Y}]^{-1}\mathbf{V}^{\mathsf{T}}\mathbf{Z}, \tag{9.12}$$

where $\mathbf{Z} = \mathbf{Y}\boldsymbol{\beta} + \boldsymbol{\xi}$. The IV estimator is a consistent estimator of $\boldsymbol{\beta}$, which can be shown by substituting $\mathbf{Z} = \mathbf{Y}\boldsymbol{\beta} + \boldsymbol{\xi}$ in (9.12):

$$\begin{aligned} E[\hat{\boldsymbol{\beta}}_{\mathrm{IV}}] &= E[[\mathbf{V}^{\mathsf{T}}\mathbf{Y}]^{-1}\mathbf{V}^{\mathsf{T}}(\mathbf{Y}\boldsymbol{\beta} + \boldsymbol{\xi})] \\ &= \boldsymbol{\beta} + E[[\mathbf{V}^{\mathsf{T}}\mathbf{Y}]^{-1}\mathbf{V}^{\mathsf{T}}\boldsymbol{\xi}]. \end{aligned}$$

For a given set of N observations $\mathbf{z} = (z_1, \ldots, z_N)$, the IV estimator can be calculated by the sample average

$$\hat{\boldsymbol{\beta}}_{\mathrm{IV}} = \boldsymbol{\beta} + [\frac{\mathbf{V}^{\mathsf{T}}\mathbf{Y}}{N}]^{-1}[\frac{\mathbf{V}^{\mathsf{T}}\boldsymbol{\xi}}{N}],$$

which converges by the weak law of large numbers to $\boldsymbol{\beta}$ w.p.1 as $\frac{\mathbf{V}^{\mathsf{T}}\mathbf{Y}}{N} \rightarrow E[\mathbf{V}^{\mathsf{T}}\mathbf{Y}]$ and $\frac{\mathbf{V}^{\mathsf{T}}\boldsymbol{\xi}}{N} \rightarrow 0$ w.p.1.

A regression with an IV transformation is called an *IV regression* (see Greene [220] and Woolrdige [581] for details and examples of IV methods) and a *generalized IV regression* if we use more than M IV variables. There are no mechanically generated IVs that work for all cases. It often requires considerable ingenuity to find good IVs and to argue that they can in fact serve to correct for endogeneity. This is what makes the technique rather difficult to apply, requiring the skills of an experienced econometrician.

In nonlinear problems, IV techniques become more difficult to apply. For example, one often encounters endogeneity when estimating discrete-choice demand models such as the MNL model from aggregate data (prices correlated with unobservable product characteristics). However, the problem is hard to correct because the aggregate demand is a nonlinear function of the utilities of each product and the endogeneity is present in the equation for the utilities. So using any IV technique for correcting for endogeneity becomes computationally challenging, as pointed out by Berry [53]. Berry [53] and Berry, Levinsohn, and Pakes [52] recommend that for the case of discrete-choice models in an oligopoly setting, one use measures of the firm's costs and the attributes of the products of the other firms as IVs. See also Besanko, Gupta, and Jain [63] for estimating a logit model in the presence of endogeneity due to competition.

9.2.5.2 Heterogeneity

Customer heterogeneity is important to understand in RM. In Section 7.2.3 we examined a few models of heterogeneity—namely, the finite-mixture logit model and the random-coefficients discrete-choice model. Here, we discuss how to estimate these models.

Estimation of the finite-mixture logit model is relatively straightforward. First, we must determine the number of segments. If there is no *a priori* knowledge of the number, we iterate the estimation procedure, increasing or decreasing the number of segments in each round, using suitable model-selection criteria (see Section 9.5.1) to decide on the optimal number of segments. For a given number of segments L, we find the parameters that maximize the log-likelihood function. For the finite-mixture logit model of Section 7.2.3.1, this would amount to maximizing the following likelihood function based on the purchase histories of N customers:

$$\mathcal{L} = \prod_{k=1}^{N} \sum_{l=1}^{L} \frac{e^{\nu_l}}{\sum_{i=1}^{L} e^{\nu_i}} \frac{e^{\beta^{l\top} \mathbf{y}_{c(k)}}}{\sum_{j=1}^{n} e^{\beta^{l\top} \mathbf{y}_j}},$$

where $c(k)$ is the choice made by customer k. The only difficulty, from an optimization point of view, is that taking logs on both sides does not convert the right-hand side into a sum of terms, so the maximization is somewhat more challenging than the estimation of standard logit models.

Estimation of the random-coefficient logit, likewise, uses maximum-likelihood estimation and is more difficult in general than the standard multinomial logit. Consider the model given in Section 7.2.3.2. Assuming the parameters follow a normal distribution, the likelihood function that needs to be maximized is given by

$$\mathcal{L} = \prod_{k=1}^{N} \int_{\boldsymbol{\beta}} \frac{e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{y}_{c(k)}}}{\sum_{j=1}^{n} e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{y}_j}} f(\boldsymbol{\beta}) d\boldsymbol{\beta}, \tag{9.13}$$

where f is the M-dimensional joint normal p.d.f. (with an identity covariance matrix if the taste parameters are independent). If the distributions of the parameters $\boldsymbol{\beta}$ are modeled as a joint normal distribution with a general covariance matrix structure, then evaluation of the integral is quite difficult in practice. However, the extreme value distribution has been integrated out in (9.13), and we do end up with a logit-like term inside the integrals.

One of the problems dealing with unobservable heterogeneity in the population is that we often have to assume a distribution of heterogeneity without having much evidence as to its specification. Many times, a distribution is chosen for analytical or computational convenience. Unfortunately, a situation can arise where two radically different distributions of heterogeneity equally support the aggregate demand observations. This was pointed out by Heckman and Singer [248], who illustrated this overparameterization with the following example:

Example 9.8 Consider an aggregate-demand function based on a heterogeneity parameter θ. The variance on the distribution of θ represents the degree of heterogeneity. Let the demand for a particular value of θ be given by the distribution

$$G_1(z|\theta) = 1 - e^{-z\theta}, \quad z \geq 0, \ \theta > 0,$$

and let θ be equal to a constant η with probability 1 (essentially saying the population is homogeneous). The aggregate-demand distribution then is $F_1(z) = 1 - e^{-z\eta}$.

Consider another possible specification where

$$G_2(z|\theta) = 1 - \int_{z(2\theta)^{-0.5}}^{\infty} \frac{2}{\sqrt{2\pi}} e^{-w^2/2} dw, \quad z \geq 0$$

and the distribution of θ given by $\eta^2 e^{-\eta^3\theta}$. This also turns out to lead to an aggregate-demand distribution given by $1 - e^{-z\eta}$. So based only on aggregate demand data, it is impossible to identify which specification is correct.

Therefore, one should proceed with caution when inferring a functional form for unobserved heterogeneity from aggregate data.

Nonparametric methods avoid the problem of having to specify a distribution, and Jain, Vilcassim, and Chintagunta [267] follow this strategy. Assume that the coefficients of the MNL model $\boldsymbol{\beta}$ in (9.10) are randomly drawn from a discrete multivariate probability distribution $G(\Theta)$. That is, the k^{th} customer is assumed to make his choice using $\boldsymbol{\beta}_k$, whose components are drawn from $G(\Theta)$. $G(\cdot)$ is considered a discrete distribution with support vectors $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L$. They estimate the number of support vectors L, the location of the support vectors, and the probability mass $\boldsymbol{\theta}_i$ associated with the i^{th} support vector from observed data.

9.2.5.3 Competition

If one has access to information on prices and demand for an entire market (such as MIDT data for airlines and scanner-panel data sold by marketing research firms), it is possible to separately estimate competitive- and own-price effects. A common strategy in such cases is to assume a model of competition between the firms, derive the equilibrium conditions implied by this model, and then estimate the parameters subject to these equilibrium conditions. We illustrate this approach with an example:

Example 9.9 Assume a homogeneous population of customers who choose among n products according to the MNL choice rule. Then the theoretical share of product j is given as in Section 7.2.2.3,

$$P_j = \frac{e^{\boldsymbol{\beta}^{\mathsf{T}} \mathbf{y}_j}}{\sum_{i=1}^{n} e^{\boldsymbol{\beta}^{\mathsf{T}} \mathbf{y}_i}}, \tag{9.14}$$

where price is one of the explanatory variables in $\mathbf{y}_j$. One way to estimate the parameters $\boldsymbol{\beta}$ is by equating the observed market share to the theoretical prediction of equilibrium. It is convenient to take logs in doing this, which yields the following system of equations relating market shares to choice behavior:

$$\ln P_j = \boldsymbol{\beta}^{\mathsf{T}} \mathbf{y}_j - \ln\left(\sum_{i=1}^{n} e^{\boldsymbol{\beta}^{\mathsf{T}} \mathbf{y}_i}\right), \quad j = 1, \ldots, n. \tag{9.15}$$

Next assume that prices are formed by a Bertrand-style competition in prices (see Section 8.4.1.4). Let c_j be the constant marginal cost of production for product j. The profit function for product j is given by

$$V_j(p_j) = (p_j - c_j) N P_j, \tag{9.16}$$

where N is the size of the population. Let β_p be the coefficient of price in (9.14). Differentiating (9.16) with respect to p_j and setting it to zero, we get the first-order equilibrium conditions,

$$(p_j - c_j)\beta_p P_j (1 - P_j) + P_j = 0, \quad j = 1, \ldots, n. \tag{9.17}$$

The vector of parameters $\boldsymbol{\beta}$ is then estimated by attempting to fit a solution to (9.15) and (9.17) simultaneously. This can be done using, say, nonlinear least-squares estimation.

9.3 Forecasting Methods

We next turn to forecasting methods, which explicitly attempt to "predict" the future values of a sequence of data. For RM, we are mostly interested in forecasting demand (demand to come, as well as aggregate demand for the resource and at various levels of aggregation), though in many cases one also needs to forecast quantities such as market prices, length of stay (in hotel RM), cancellation and no-show rates, and so on. Indeed, the methods presented here, by and large, apply to a wide variety of forecasting tasks, though for purposes of illustration we focus on demand forecasting as our canonical application.

Forecasting is a vast topic, spanning a diverse range of fields including statistics, computer science, engineering, and economics. Over the years, a core set of forecasting methods have been developed and new improvements continue despite the maturity of the field. Some of these forecasting methods are based on rigorous mathematical and statistical foundations, while others are largely heuristic in nature.

Yet despite this long history and vast body of research on forecasting, there are few published reports that document the performance of various forecasting methods in RM applications. Presentations on forecasting by practitioners at industry conferences often suffer from the proprietary nature of the material, with key details either omitted or disguised. The same can be said of most presentations by RM system vendors. Nevertheless, one can still glean some useful insights into current practice from these sources.

For one, most forecasting algorithms in RM practice are variations of standard methods, and most are not particularly complicated or mathematically sophisticated. Also, many vendors use multiple algorithms, which allow users the option of choosing one or more methods, or, alternatively, the system may combine the forecasts from the various methods itself (see Section 9.3.11). Finally, the majority of forecasting effort in practice is directed at data-related tasks—collection, preprocessing and cleansing—rather than on forecasting methodology per se.

In terms of forecasting methods, the emphasis in RM systems is on speed, simplicity, and robustness, as a large number of forecasts have to be made and the time available for making them is limited. For example, if an airline has 50,000 itineraries in 10 fare classes that it reforecasts 40 times during a sales period (typical numbers for a medium-size airline), then they must forecast nearly *2 million* demand quantities every day! And this does not include forecasts of important auxiliary quantities such as cancellation and no-show rates. It is little wonder, then, that fast, simple methods are preferred in RM systems.

Forecasting is normally performed overnight in a batch process and then fed to the optimization modules, so the time window for completing all control operations ranges from six to eight hours at most. Forecasting model calibration (estimation), in turn, can only be done off line and infrequently.

Robustness of the forecasts is also important in practice for these same reasons. If a large number of forecasts are off widely and the system starts generating exceptions, analysts may be overwhelmed by the amount of manual intervention required. Hence, performance—in terms of forecast accuracy under "normal" data conditions—while always a desirable criteria, has to be balanced against these "real world" speed constraints and robustness considerations. We next provide an overview of RM forecasting algorithms, starting with ad hoc and time-series methods and progressing to Bayesian, state-space (Kalman filter), and machine-learning (neural network) methods.

9.3.1 Ad-Hoc Forecasting Methods

The first-class of methods we look at are known as ad-hoc forecasting methods because their reasoning is largely heuristic in nature. The term *ad hoc* is somewhat misleading, however, as many of these methods turn out to have good theoretical properties despite their heuristic origins. They are also sometimes referred to as *structural* forecasting methods because they proceed by assuming a compositional structure on the data, breaking up and composing the series into hypothesized patterns (see Figure 9.3). These include the following three types of components:

- **Level** The typical or "average" value of the data, though in ad-hoc methods the level is not defined as a statistical average in any formal sense.

- **Trend** A predictable increase or decrease in the data values over time. Most often these are modeled as linear increases or decreases, but other functions may be used.

- **Seasonality** A periodic or repeating pattern in the data values over time—for example, as produced by day-of-week or time-of-year effects.

Ad-hoc forecasting methods are intuitive, are simple to program, and maintain and perform well in practice. For these reasons, they are prevalent in RM practice.

A common strategy of ad-hoc forecasting methods is to try to "smooth" the data or average-out the noise components to estimate the level, trend, and seasonality components in the data. These estimates of

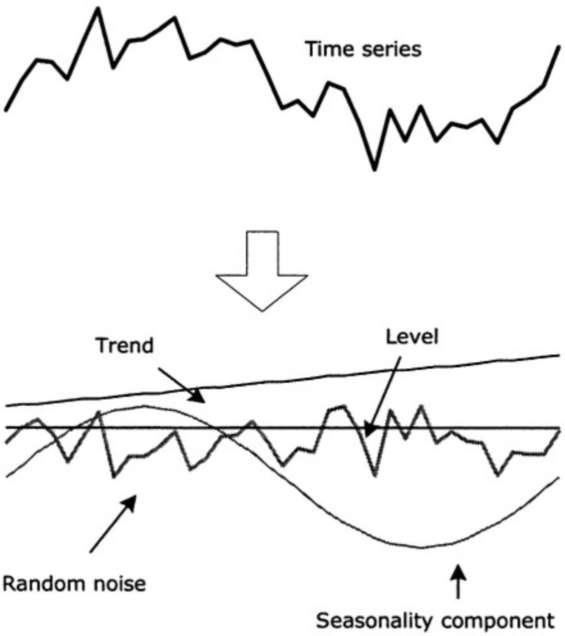


Figure 9.3. Time series of data and its components.

the smoothed series are then used to forecast future values, as we show next.

9.3.1.1 *M*-Period Moving Average

Let t represent the current time, and suppose we want to forecast values at time $t+k$ in thefuture, $\hat{Z}_{t+k}$, called the *k-period ahead forecast.* Let $z_1, \ldots, z_t$ denote the observed demand data, and $\hat{Z}_{t+1}, \ldots, \hat{Z}_{t+K}$ denote the forecasts. To forecast one period ahead, one simple approach is to use the average of the past M observations. That is, the forecast for period $t+1$ is given by

$$\hat{Z}_{t+1} = \frac{z_t + z_{t-1} \ldots + z_{t-M+1}}{M}, \tag{9.18}$$

called the *simple M-period moving-average forecast. M* is called the *span* of the moving average. The formula for the k-period ahead forecast is given by

$$\hat{Z}_{t+k} = \hat{Z}_{t+1}, \quad k = 2, \ldots, K.$$

A different way of writing (9.18) is

$$\hat{Z}_{t+1} = \hat{Z}_t + \frac{z_t - z_{t-M}}{M},$$

which is computationally faster. If t is less than M (that is, in the initial stages of forecasting), one can use $M = t$.

The moving-average method is very simple and fast, but its motivation is largely heuristic. The idea is simply that the most recent observations serve as better predictors for the future than do older data. Therefore, instead of taking the forecast as the average of *all* the data, we average only the M most recent data observations.

The moving-average forecast responds more quickly to underlying shifts in the demand process if the span M is small, but a small span results in a more volatile forecast (one that is more sensitive to noise in the data). In practice, M may range from 3 to 15, but the value depends heavily on the data characteristics and the units used for the time intervals.

When the data exhibits an upward or downward trend, the moving average method will systematically underforecast or overforecast. To handle such cases, variations such as double or triple moving average have been developed, but for such data one of the exponential smoothing methods given next is usually preferred.

9.3.1.2 Exponential Smoothing

Exponential-smoothing methods are among the most popular forecasting methods used in RM practice because they are simple and robust and generally have good forecast accuracy. We look at three variations of exponential smoothing. First, however, we formally define the following component estimates of the forecast:

$$\begin{aligned} A_t &= \text{the estimate of the level (average) for period } t, \\ T_t &= \text{the estimate of the trend for period } t, \text{ and} \\ S_t &= \text{the estimate of the seasonality factor for period } t. \end{aligned}$$

See Figure 9.3 for an illustration of these components.

Simple Exponential Smoothing This simplest version of exponential smoothing is defined by a single parameter, $0 < \alpha < 1$, called the *smoothing constant for the level.* The forecast for time-period $t + 1$ is given by

$$\hat{Z}_{t+1} = A_t = \alpha z_t + (1 - \alpha)\hat{Z}_t. \tag{9.19}$$

The k-period ahead forecast is then simply

$$\hat{Z}_{t+k} = \hat{Z}_{t+1}, \; k = 1, \ldots, K.$$

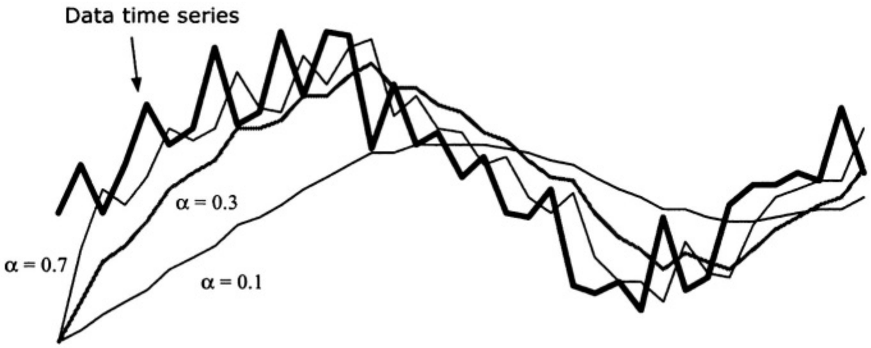


Figure 9.4. **Exponential smoothing with different smoothing parameters α.**

The choice of α is a design decision and is usually calibrated prior to starting the forecasting system. Smaller values of α smooth the forecasts more, leading to more stability, while larger values of α make the forecast more responsive to recent changes in level but also more susceptible to noise. In practice, α is typically set between 0.05 and 0.3 in RM applications. In addition, more advanced adaptive variations of the smoothing methods attempt to automatically optimize the value of α based on its observed performance.

Some motivation for the exponential smoothing method can be obtained by expanding the recursive formula (9.19), substituting repeatedly for $\hat{Z}_{(\cdot)}$:

$$\begin{aligned}
\hat{Z}_{t+1} &= \alpha z_t + (1-\alpha)\hat{Z}_t \\
&= \alpha z_t + (1-\alpha)(\alpha z_{t-1} + (1-\alpha)\hat{Z}_{t-1}) \\
&= \alpha z_t + \alpha(1-\alpha) z_{t-1} + (1-\alpha)^2 \hat{Z}_{t-1} \\
&= \vdots \\
&= \alpha \sum_{j=0}^{\infty} (1-\alpha)^j z_{t-j}. \qquad (9.20)
\end{aligned}$$

Thus, we see the forecast for period $t+1$ is a weighted combination of all previous observations with the weights "exponentially" decreasing at a rate of $(1-\alpha)$. High values of α make the decrease rapid, and the forecasts will be more responsive to recent observations, while low values of α will spread the weights over a longer period, and the forecasts will react more slowly to changes in demand. Figure 9.4 illustrates the role of the smoothing parameter on a sample time series of data.

The smoothing parameters play a similar (albeit more complicated) role in the next two models.

Exponential Smoothing with Linear Trend Let $0 < \alpha < 1$ and $0 < \beta < 1$ be two parameters representing the smoothing factors for the underlying level and trend, respectively. Then the forecast for time-period $t+1$, $\hat{Z}_{t+1}$ is given by the following formulas:

$$\hat{Z}_{t+1} = A_t + T_t, \tag{9.21a}$$
$$A_t = \alpha z_t + (1-\alpha)(\hat{Z}_t + T_t) \tag{9.21b}$$
$$T_t = \beta(\hat{Z}_t - \hat{Z}_{t-1}) + (1-\beta)T_{t-1}. \tag{9.21c}$$

The k-period ahead forecast is given by

$$\hat{Z}_{t+k} = A_t + kT_t, \;\; k = 1, \ldots, K.$$

Note T_t is the estimate of the trend factor in each period and is smoothed using β.

Exponential Smoothing with Trend and Seasonality (Holt-Winter's Method) This method is applicable to data series that exhibit seasonal variations (for example, monthly, quarterly, or half-yearly variations). Let $0 < \alpha < 1$, $0 < \beta < 1$, and $0 < \gamma < 1$ be three parameters used to control the smoothing on the underlying level, trend, and seasonality, respectively. Let L represent the periodicity of the seasonality—that is, the number of periods after which the seasons repeat. L depends on the length of the periods and the seasonality—for instance, if we are constructing quarterly forecasts and the seasonality is by quarter, $L = 4$, or if we are constructing monthly forecasts and the seasonality is by month, $L = 12$. Then the forecast for time-period $t+k$ is given by the formula,

$$\hat{Z}_{t+k} = (A_t + kT_t)S_{t+k-L}, \;\; k = 1, \ldots, K, \tag{9.22}$$

and the three components of this forecast are updated as follows:

$$A_t = \alpha\left(\frac{z_t}{S_{t-L}}\right) + (1-\alpha)(\hat{Z}_t + T_t) \tag{9.23a}$$
$$T_t = \beta(\hat{Z}_t - \hat{Z}_{t-1}) + (1-\beta)T_{t-1} \tag{9.23b}$$
$$S_t = \gamma\left(\frac{z_t}{S_t}\right) + (1-\gamma)S_{t-L}. \tag{9.23c}$$

In (9.23c), S_t is the new estimate of the seasonality factor for period t. These factors are updated once each season and are smoothed with the previous estimate of the seasonality factor, of L periods in the past, using γ. Equation (9.23a) "deseasonalizes" the data by replacing z_t by $\frac{z_t}{S_{t-L}}$ and then updates this deseasonalized data using the same procedure as in exponential smoothing with a linear trend.

The deseasonalized forecast is "reseasonalized" in (9.22) by multiplying by the estimated seasonality factor S_{t+k-L} to generate the forecast $\hat{Z}_{t+k}$. More than one seasonal factor can be incorporated into the model (such as both a day-of-week factor and a monthly factor) by simply keeping two multiplicative seasonal factors and updating them as in (9.23a) and (9.23c).

9.3.2 Time-Series Forecasting Methods

In contrast to ad-hoc forecasting methods, time-series methods are based on well-specified classes of models that describe the underlying time series of data. These models have relatively simple mathematical structures, yet the model classes are rich enough to represent a wide range of data characteristics. Since the models are well specified, it is possible to derive "optimal" (MSE or ML) forecasting methods for each one. In this way, the forecasting procedure is specifically tailored to the underlying data-generation model. This formal representation of the dynamics governing the time series and the rigorous development of optimal forecasting methods is what distinguishes time-series methods.

The collection of random variables $\{Z_t\}$ is called a *time series* if it represents successive observations taken over time. The values Z_t are assumed to be generated by a dynamic system, which may depend on past values Z_s for $s \leq t$ and a series of random disturbances $\{\xi_t\}$. At time t, we have observations of the past data values $z_t, z_{t-1}, \ldots$ and would like to forecast the future values of the time series—for example, forecasting the value k units in the future, or Z_{t+k}. We might be interested in a single point estimate, $\hat{Z}_{t+k}$, of this future value Z_{t+k} or an estimate of the parameters of its distribution.

A time-series forecasting process proceeds in two basic steps. First, we make a hypothesis about the specific type of process generating the time series of data. Various model-identification techniques can be employed to help determine which models best fit the data. Once the model is identified, we estimate its parameters. Finally, we apply the corresponding optimal forecasting method specific to that model.

One distinct advantage of time-series methods is that they explicitly model the correlations between successive data points and exploit any dependence to make better forecasts. However, it is up to the RM system designer to decide if such correlations exist (for example, whether there are "runs" in the data, where high-demand observations are often followed by other high-demand observations). Moreover, even when correlation exists, the designer must decide if it is worth building in this extra complexity to obtain better forecasts because these models require relatively large samples of data (usually at least 50 observations) to calibrate accurately.

In what follows, we present several time-series models and methods for forecasting and updating estimates of their parameters. But first, we introduce two important concepts central to time-series forecasting: stationarity and autocorrelation.

9.3.2.1 Stationary Time Series

Stationarity is an important property of a time series that greatly simplifies the forecasting task. Simply put, a time series is stationary if its statistical properties do not change over time. More formally, if $Z_t, \ldots, Z_{t+k}$ and $Z_{t+m}, \ldots, Z_{t+k+m}$ are two sets of k random variables from the series, then the series is said to be *stationary* if the joint distribution of these two sets of variables is the same for all choices of time t and all pairs of values k and m.

To understand why the stationarity assumption simplifies forecasting, consider the problem of estimating the first two moments (means, variances, and covariances) of a collection of *N* random variables from a nonstationary time series. Nonstationarity means that these *N* observations were generated by a random process whose joint distribution could be different at each time. Therefore, to estimate the first and second moments, we need to estimate *N* expected values, *N* variances, and *N*(*N* – 1)/2 covariances—a total of $N^2/2+3N/2$ parameters. However, if the series is stationary, all the expected values and variances will be the same, as Z_t and Z_{t+k} have the same marginal distribution. Moreover, there are only *N* – 1 distinct covariances because the joint distribution of Z_t and Z_{t+k} is the same as that of Z_{t+m} and Z_{t+k+m} (for all t, k and m), and hence their respective covariances are the same. Therefore, the number of parameters we need to estimate if the series is stationary is only 2 + (*N* – 1), a much more manageable task. To simplify things even further, one often makes further structural assumptions that guarantee that a large number of the covariances are identically zero, making the estimation problem even simpler.

How serious is the assumption of stationarity? At first glance, it seems quite restrictive. In fact, many time series encountered in practice are clearly nonstationary. For example, any time-series data with a trend or seasonal pattern is not stationary (if the series shows an increasing trend, the underlying distributions of the successive random variables are certainly not identical). However, even if the time series itself is not stationary, transformations of the series—such as the difference between successive values—may be stationary. Indeed, time-series forecasting methods for nonstationary data typically involve transforming the data to obtain related stationary series; forecasts based on this transformed stationary series are then used construct a forecast for the original time series.

9.3.2.2 Autocorrelation

As we show below, entire classes of stationary time-series methods are specified through their covariance structure over time—that is, the covariance of Z_t and Z_{t+k} for all k. The autocorrelation function (ACF) and partial autocorrelation function (PACF) are the key tools to analyze this covariance structure. They serve as "signatures", as it were, of a time-series model, and by comparing these signatures to the "sample" signatures obtained from our data we can determine which models are most appropriate.

Specifically, the j^{th} *autocovariance* function is defined as the covariance between Z_t and Z_{t+j}:

$$\gamma_j = \operatorname{Cov}(Z_t, Z_{t+j}).$$

The autocovariance function measures the dispersion or variance of the process. However, two data series that are identical except for the scale of measurement will have different autocovariance functions. Therefore, it is better to deal with the *autocorrelation* function, defined as the autocovariance function divided by the variance

$$\rho_j = \frac{\gamma_j}{\gamma_0},$$

which is scale invariant.

Given a data series $z_1, \ldots, z_N$, the j^{th} *sample autocovariance* function is given by

$$c_j = \frac{\sum_{t=1}^{N-j}(z_t - \bar{z})(z_{t+j} - \bar{z})}{N},$$

where $\bar{z}$ is the sample mean

$$\bar{z} = \frac{1}{N}\sum_{t=1}^{N} z_j.$$

The j^{th} *sample autocorrelation function* is given by

$$r_j = \frac{c_j}{c_0}.$$

The partial autocorrelation function (PACF) is defined as

$$\operatorname{Corr}(Z_t, Z_{t+j} | Z_{t+1}, \ldots, Z_{t+j-1})$$

and can be shown to be equal to the ratio of two determinants involving the autocorrelations (see Wei [560], pp.15–22). A sample PACF can be defined analogous to the sample ACF, but it is considerably more complex to compute. However, most statistical packages automatically compute and plot the sample ACFs and PACFs, so the complexity of the calculations is not a major concern. An example of a sample autocorrelation function and a partial autocorrelation function is shown in Figure 9.5.

9.3.3 Stationary Time-Series Models

We first consider stationary time-series models. To begin, define a *linear filter* as a stochastic process $\{Z_t\}$ that can be written as an infinite weighted sum of random variables as follows:

$$Z_t = \mu + \xi_t - \psi_1 \xi_{t-1} - \psi_2 \xi_{t-2} - \cdots, \tag{9.24}$$

(the minus sign on the ψ's is by convention), where ψ_t and μ are constant parameters and the random variables ξ_t (called *white-noise disturbances*) are assumed to be i.i.d. normally-distributed random variables with a mean of 0 and standard deviation σ_ξ for all t. The stochastic process $\{\xi_t\}$ is therefore a stationary process. We define μ to be the level of the series, which is assumed to be constant. If the sequence $\psi_1, \psi_2, \ldots$ is finite or is infinite and convergent, then one can show that the process $\{Z_t\}$ is stationary and μ is the mean of the series ($E[Z_t] = \mu$).

We can rewrite equation (9.24) to express Z_t in terms of $Z_{t-1}, Z_{t-2}, \ldots$, and ξ_t as follows:

- First, eliminate ξ_{t-1} from (9.24), and write ξ_t, ξ_{t-1} in terms of the remaining variables and parameters,

$$\xi_t = Z_t - \mu + \psi_1 \xi_{t-1} + \psi_2 \xi_{t-2} + \cdots \tag{9.25}$$

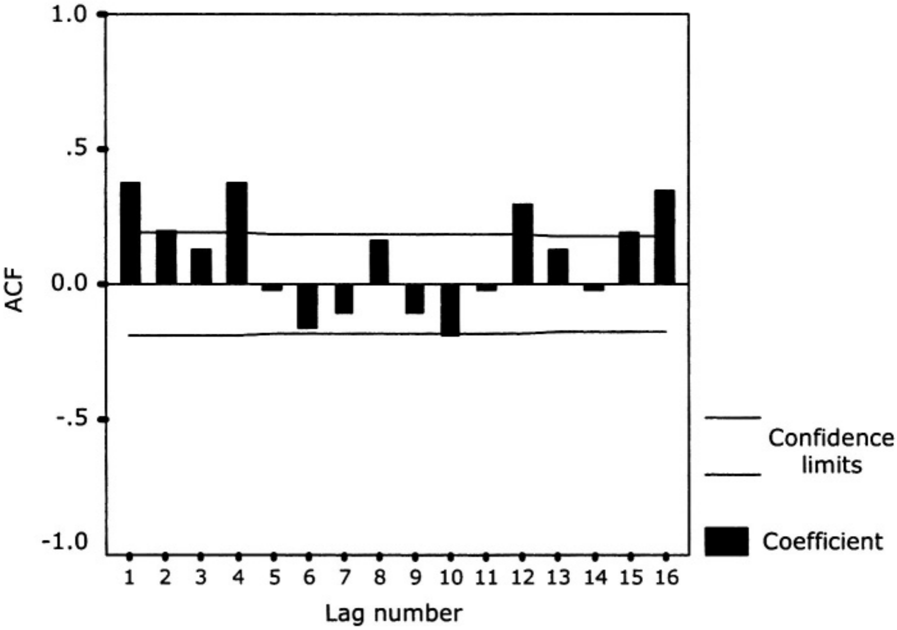
$$\xi_{t-1} = Z_{t-1} - \mu + \psi_1 \xi_{t-2} + \psi_2 \xi_{t-3} + \cdots . \tag{9.26}$$

- Substitute (9.26) in (9.25) to obtain

$$\begin{aligned} Z_t = {} & \mu(1 + \psi_1) - \psi_1 Z_{t-1} + \xi_t + (-\psi_1^2 - \psi_2)\xi_{t-2} \\ & +(-\psi_1\psi_2 - \psi_3)\xi_{t-3} + \cdots . \end{aligned} \tag{9.27}$$

- Repeat this process to eliminate ξ_{t-2}, ξ_{t-3} and so on to obtain an equation where Z_t is expressed solely in terms of $Z_{t-1}, Z_{t-2}, \ldots$, and ξ_t:

$$Z_t = \delta + \xi_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots, \tag{9.28}$$



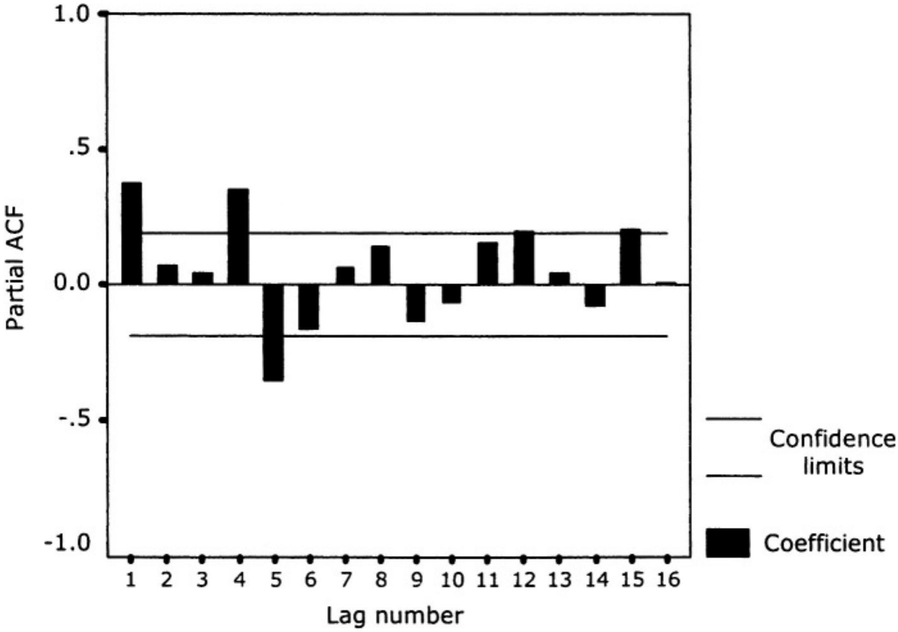


Figure 9.5. Plots of sample autocorrelation function and a partial autocorrelation function.

where δ and $\theta_1, \theta_2, \ldots$ are a new set of constants that depend on μ and $\psi_1, \psi_2, \ldots$.

The random variable Z_t can represent either a stationary or nonstationary process depending on the properties of the parameters θ_i (equivalently, ψ_i). Three important stationary time-series models arise from using (9.24) and (9.28):

- **Moving average process (MA(q))** This process requires that only a finite number of ψ's be nonzero in (9.24). A q^{th} order MA process is given by

$$Z_t = \mu + \xi_t - \psi_1 \xi_{t-1} - \cdots - \psi_q \xi_{t-q}. \tag{9.29}$$

- **Autoregressive process (AR(p))** This process requires that only a finite number of θ's be nonzero in (9.28):

$$Z_t = \delta + \xi_t + \theta_1 Z_{t-1} + \cdots + \theta_p Z_{t-p}. \tag{9.30}$$

- **Autoregressive moving average process (ARMA(p, q))** This process is a combination of MA and AR process

$$\begin{aligned} Z_t = {} & \delta + \xi_t + \theta_1 Z_{t-1} + \cdots + \theta_p Z_{t-p} \\ & -\psi_1 \xi_{t-1} + \cdots + \psi_q \xi_{t-q}. \end{aligned} \tag{9.31}$$

An AR process is stationary if the roots of the polynomial $1 + \theta_1 x + \cdots + \theta_p x^p$ are greater than one. An MA process is called *invertible* if all the roots of the polynomial $1 + \psi_1 x + \cdots + \psi_q x^q$ are greater than one. One can show that a finite-order stationary AR process can be expressed as an infinite-order MA process, and conversely, a finite-order invertible MA process can be written as an infinite-order AR process. This relationship is useful because if a fitted AR model contains a large number of parameters, it is possible that the corresponding MA model will have fewer parameters, and vice versa. An ARMA model, being a combination of an AR and an MA process can, in principle, reduce the number of parameters even further. Every ARMA(p, q) model has what is called a *pure MA representation*—that is, it can be written as the following infinite sum (see Wei [560], p.58 for a derivation):

$$Z_t = \mu + \xi_t - \psi_1 \xi_{t-1} - \cdots - \psi_q \xi_{t-q} - \cdots.$$

In most practical applications of these models, p and q rarely exceed 2. The means and covariances for ARMA series with small values of p and q are given in Table 9.2. Recall γ_k denotes the covariance

Table 9.2. **Means and covariances of some stationary time-series processes.**

Process	Z_t	*Mean*	*Autocovariances* (γ_k)
AR(1)	$\delta+\theta_1 Z_{t-1}+\xi_t$	$\frac{\delta}{1-\theta_1}$	$\theta_1^k\frac{\sigma_\xi^2}{1-\theta_1^2}$
AR(2)	$\delta+\theta_1 Z_{t-1}$ $+\theta_2 Z_{t-2}+\xi_t$	$\frac{\delta}{1-\theta_1-\theta_2}$	$\theta_1\gamma_{k-1}+\theta_2\gamma_{k-2}+\sigma_\xi^2 \quad k=0$ $\theta_1\gamma_{k-1}+\theta_2\gamma_{k-2} \quad k>0$
MA(1)	$\mu-\psi_1\xi_{t-1}+\xi_t$	μ	$(1+\psi_1^2)\sigma_\xi^2 \quad k=0$ $-\psi_1\sigma_\xi^2 \quad k=1$ $0 \quad k\geq 2$
MA(2)	$\mu-\psi_1\xi_{t-1}$ $-\psi_2\xi_{t-2}+\xi_t$	μ	$(1+\psi_1^2+\psi_2^2)\sigma_\xi^2 \quad k=0$ $-\psi_1(1-\psi_2)\sigma_\xi^2 \quad k=1$ $-\psi_2\sigma_\xi^2 \quad k=2$ $0 \quad k\geq 3$
ARMA(1, 1)	$\delta+\theta_1 Z_{t-1}$ $-\psi_1\xi_{t-1}+\xi_t$	$\frac{\delta}{1-\theta_1}$	$\frac{1+\psi_1^2-2\theta_1\psi_1}{1-\theta_1^2}\sigma_\xi^2 \quad k=0$ $\frac{(\theta_1-\psi_1)(1-\theta_1\psi_1)}{1-\theta_1^2}\sigma_\xi^2 \quad k=1$ $\theta_1\gamma_{k-1} \quad k\geq 2$

$E[Z_t Z_{t+k}] = E[Z_t Z_{t-k}]$. Note that for stationary processes, the covariances are independent of t, with γ_0 representing the variance. In some cases (as for AR(2)), the covariances do not have a closed-form formula but can be derived as solutions to a set of equations (see Wei [560] for derivations). The AR and MA processes have distinctive ACF and PACFs. Figure 9.6 shows some typical theoretical ACF and PACFs. The forms of these ACF and PACFs provide important clues as to which model is most appropriate for the observed data. Such model-identification issues are discussed in Section 9.3.5.

Once we decide that a set of time-series data is an $\mathrm{MA}(q)$ or an $\mathrm{AR}(p)$ process, we can proceed to identify the parameters of the model by using ML or MSE criteria. We can then use the models for forecasting in a relatively straightforward manner, as shown in the following example.

Example 9.10 We illustrate the forecasting process on the following data set

$$\{z_t\}_{t=1,\ldots,24} = \{25.11, 17.23, 17.87, 17.80, 17.49, 17.99, 18.59, 19.08, 19.55, 19.50, 20.74, 21.32, 20.76, 21.10, 21.03, 21.75, 21.17, 19.01, 18.95, 17.75, 17.22, 16.52, 17.35, 16.61\}.$$

Assume the data comes from an AR(2) process,

$$Z_t = \delta+\xi_t+\theta_1 Z_{t-1}+\theta_2 Z_{t-2}. \tag{9.32}$$

The forecasting process proceeds as follows:

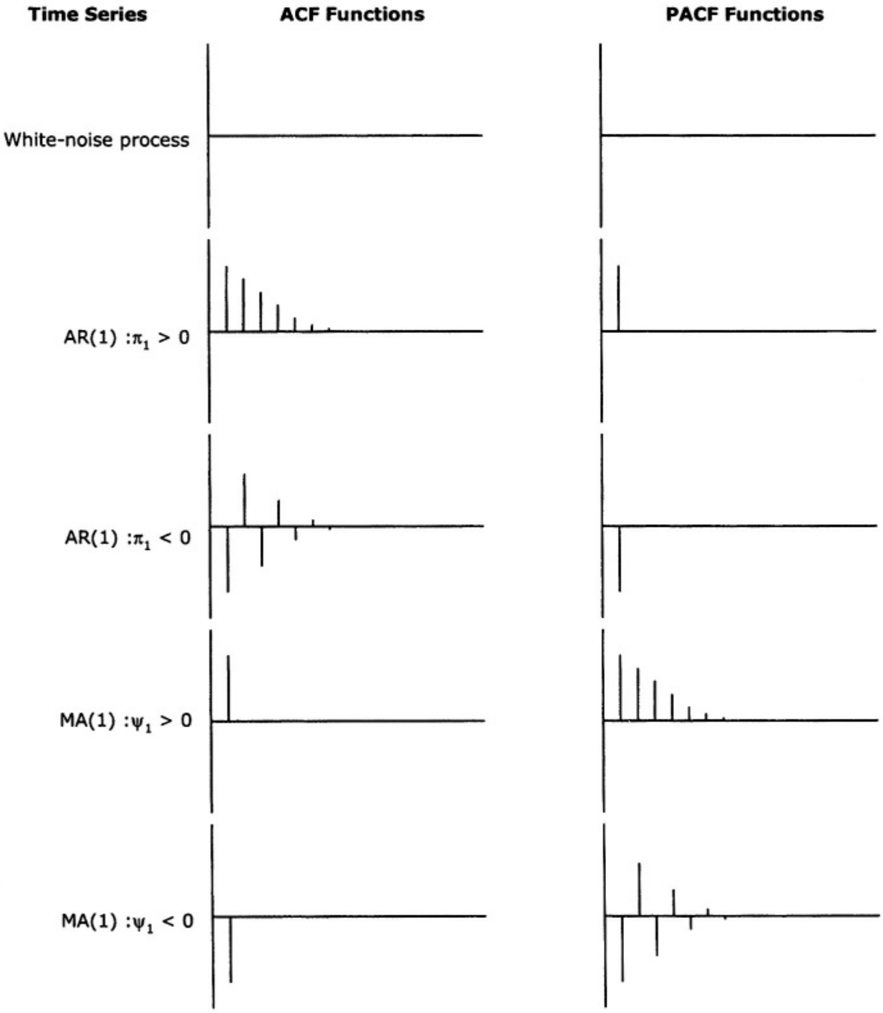


Figure 9.6. The ACFs and PACFs for a few simple stationary time-series models.

Parameter estimation We first estimate the parameters δ, θ_1, and θ_2 in (9.32) by MSE estimation. This we do by solving the following optimization problem (note that an AR(2) process requires at least two initial points, so we begin with data point 3):[3]

$$\min_{\delta,\theta_1,\theta_2} \sum_{k=3}^{t} (z_k - (\delta + \theta_1 z_{k-1} + \theta_2 z_{k-2}))^2 \tag{9.33}$$

Let $\hat{\delta}$, $\hat{\theta}_1$, and $\hat{\theta}_2$ denote the parameters that minimize the mean-square error on these data. For an AR(2) process, (9.33) has a closed-form solution, but in general numerical optimization is required to find the minimum. (Most statistical software packages solve this optimization problem automatically.) The parameters that minimize the mean-square error for the data set (9.10) turn out to be $\hat{\delta} = 1.5, \hat{\theta}_1 = 0.891$, and $\hat{\theta}_2 = 0.0285$.

Forecast For an AR(2) process, the one-step forecast depends on the two previous observations. In general, the k-period forecasts $\hat{Z}_{t+k}$ for $k = 1, 2, 3, \ldots$ are then given by (assume $t > 2$)

$$\begin{aligned} k = 1: &\quad \hat{Z}_{t+1} = 1.5 + 0.891 z_t + 0.0285 z_{t-1} \\ k = 2: &\quad \hat{Z}_{t+2} = 1.5 + 0.891 \hat{Z}_{t+1} + 0.0285 z_t \\ k \geq 3: &\quad \hat{Z}_{t+l} = 1.5 + 0.891 \hat{Z}_{t+k-1} + 0.0285 \hat{Z}_{t+k-2}. \end{aligned} \tag{9.34}$$

The results of the forecast are given in Table 9.3.

9.3.4 Nonstationary Time-Series Models

As mentioned, most time-series data encountered in practice are nonstationary. In such cases, stationary time-series models may not fit the data well and can produce poor forecasts. Techniques to deal with nonstationary data try to make the data stationary by a suitable transformation, so that one can then apply a stationary time-series model to the transformed data. The resulting stationary forecasts are then transformed back to their original nonstationary form. Differencing successive points in the time series is one such technique.

Time series that are stationary after successive differencing are called *homogenous* nonstationary series. This means that after differencing the series is adequately represented by an ARMA model of the form (9.31). Other transformations, such as taking the logarithm of the series, can make a series stationary if the relative or percentage changes are stationary rather than the differences. For ease of exposition, however, we focus only on differencing in this section.

Given a time series Z_t, define a new time series W_t as $W_t = Z_t - Z_{t-1}$. W_t is called the first-difference of the Z_t series. As we mentioned, there is often good reason to suspect that even if Z_t is not stationary, W_t might be. A series with a linear trend, for instance, has constant differences and

[3] Solving the minimization problem (9.33) could be computationally quite intensive, especially if it has to be re-solved after each observation. A practical alternative is to estimate parameters only periodically—say, after every 50 observations.

Table 9.3. Results of the AR(2) forecasting example.

Period	*Data*	*Forecast*	*Error*	*MAD*
1	25.11			
2	17.23			
3	17.87	17.57	0.30	0.30
4	17.80	17.91	-0.11	0.21
5	17.49	17.87	-0.37	0.26
6	17.99	17.59	0.40	0.30
7	18.59	18.03	0.56	0.35
8	19.08	18.57	0.51	0.38
9	19.55	19.03	0.51	0.40
10	19.50	19.46	0.05	0.35
11	20.74	19.43	1.31	0.46
12	21.32	20.53	0.79	0.49
13	20.76	21.08	-0.32	0.48
14	21.10	20.61	0.50	0.48
15	21.03	20.89	0.14	0.45
16	21.75	20.84	0.92	0.48
17	21.17	21.48	-0.31	0.47
18	19.01	20.98	-1.97	0.57
19	18.95	19.04	-0.09	0.54
20	17.75	18.92	-1.17	0.57
21	17.22	17.86	-0.64	0.58
22	16.52	17.35	-0.82	0.59
23	17.35	16.71	0.64	0.59
24	16.61	17.43	-0.82	0.60

its first-difference series would be stationary. If W_t still is not stationary, we can construct a new series that is the differences of W_t and examine if it is stationary, and so on.

An *autoregressive integrated moving-average process,* $\text{ARIMA}(p, d, q)$, is one whose d^{th} differenced series is an $\text{ARMA}(p, q)$ process. As for the case of ARMA models, the parameters p, d, q are usually small (less than or equal to 2) in real-world forecasting models.

How do we decide how many differences to take or whether to difference at all? The ACF is helpful in this regard. If the series is nonstationary, the sample ACF shows high values for many periods, whereas if the series is stationary, it damps down to zero quickly, often within four or five periods. We can then difference the data and analyze the resulting ACF to see if the results indicate stationarity. If not, then more differencing may be needed.

The $\text{ARIMA}(p, d, q)$ model is designed for homogeneous, nonstationary time series. For example, when there is a trend (linear or nonlinear), then successive differencing of ARIMA converts the series to a stationary series. If, however, the data has a seasonal pattern in addition to a trend, a more involved procedure is required. One option is to consider the series as a product of two stationary series—one that represents the seasonal component and another that represents a stationary time series. We can then difference the seasonal component by the period of seasonality, and the other component can be treated as a stationary time series. However, model identification, parameter estimation, and forecasting are considerably more complicated for this sort of model and are beyond the scope of this chapter.

Finally, we note there is a heuristic relationship between ARIMA process and the simple exponential smoothing method (9.3.1.2). To see this, consider the following ARIMA(0,1,1) series:

$$Z_{t+1} - Z_t \;=\; \xi_{t+1} - \psi\xi_t \tag{9.35}$$

and

$$\xi_t \;=\; Z_t - Z_{t-1} + \psi\xi_{t-1}. \tag{9.36}$$

Substituting successively for $\xi_t, \xi_{t-1}, \ldots$ in the form (9.36) into (9.35), we obtain

$$Z_{t+1} \;=\; \xi_{t+1} + (1-\psi)\sum_{j=0}^{\infty} \psi^j Z_{t-j}.$$

Note the similarity with the simple exponential smoothing method (9.20), where $\alpha = (1-\psi)$. Box and Jenkins [85] and Harvey [243] derive many connections like this between ad-hoc models and ARIMA models.

9.3.5 Box-Jenkins Identification Process

Determining the model that best represents a given time series is more of an art than a science. Often many different models must be tried before one can narrow down the choice of a "best" model. However, the Box-Jenkins method provides a framework to formalize the model-selection process. It recommends an iterative methodology of choosing the model, validating it, and modifying it to identify the best possible time-series model. Here, we briefly review this methodology.

The first step in the process is *identification.* In this step, the sample ACF and PACF functions are plotted to tentatively identify the most likely candidate for a model. These correlograms are then compared with the correlograms of a standard process such as $\text{MA}(q)$, $\text{AR}(p)$, or $\text{ARMA}(p,q)$ for small values of p and q. For instance, if the sample ACF stops after q spikes, an $\text{MA}(q)$ model would be appropriate; if the sample PACF stops after p spikes, an $\mathrm{AR}(p)$ model would be appropriate; if neither looks like the right model but the correlograms still decline exponentially toward zero, an ARMA model would be more suitable.

The next step is an *estimation* step in which the model parameters are estimated from the data. Usually, these are least-squares or maximum-likelihood estimates.

The final step is the *diagnostic* step, which verifies that the chosen model and parameters indeed fit the data well. We do this by taking the ACF of the residual series (actual data values subtracted from the model prediction data) and performing various statistical tests (such as the Box-Pierce test) to see if it represents white noise. If the model performs poorly on these tests, the model is rejected, and another model is tested.

Once the model has been selected, we can then use it to generate forecasts as illustrated in Example 9.10. In practice, once a system is operational, the model itself is rarely altered. In contrast, nonparametric or semiparametric methods, such as neural-network methods, adapt the model automatically based on recently observed data. Indeed, the substantial amount of manual work and statistical skills required to implement the Box-Jenkins methodology are its main disadvantages in practice, especially in a RM context where one needs a highly automated forecasting system with minimum manual intervention. As a result, time-series methods have not found much favor in current RM practice. But their performance, when sufficiently tuned and calibrated, can be significantly better than the simpler ad-hoc forecasting methods of Section 9.3.1. So even if they are not used operationally, time-series methods play an important role as reference methods when evaluating simpler forecasting methods.

9.3.6 Bayesian Forecasting Methods

Bayesian methods are a large class of forecasting methods that use the Bayes formula to merge a prior belief about forecast values with information obtained from observed data. The methods are especially useful when there is no historical data, a common occurrence when new products are introduced. For example, an airline may start flying on a new route and have no historical demand information on the route. Fashion apparel products often change every season, and hence demand may be unrelated to the historical sales of past products. Similarly, a TV broadcaster has no historical demand information on demand for a new series. Nevertheless, in each of these cases forecasters may have some subjective beliefs about demand, based on human judgment or alternative data sources (such as test marketing and focus groups). Bayesian methods provide a rigorous and systematic way of specifying such prior beliefs and then updating them as demand data is observed. Hence, they make it possible to combine subjective knowledge with information obtained from data and observations.

9.3.6.1 Basic Bayesian Forecasting

As before, let $Z_1, Z_2, \ldots$ be a sequence of i.i.d. random variables representing a data-generation process. We assume Z_t has a density function $f(z|\theta)$ that is a function of a single, unknown parameter θ. For example, Z_t might have a Poisson distribution, and the parameter θ might be the mean λ. Since θ is unknown, it too is assumed to be a random variable with a probability density $g(\theta)$. This density, called the *prior,* represents our current belief about the value of the parameter θ. Roughly, if we are confident about the value of θ, then the density $g(\theta)$ would be tightly concentrated (have a low variance); conversely, if we are very unsure about the value of θ, then it would be more spread out (have a higher variance). A prior with a large variance is called a *diffuse* prior.

When new data is observed, we may change our belief about the parameter θ. The procedure for formalizing this updating is given by Bayes rule. Let $g_0(\theta)$ represent our initial ($t = 0$) prior distribution and z_1 denote our first observation. Then after observing demand, our *posterior distribution* of θ is given by

$$g_1(\theta) = \frac{g_0(\theta)f(z_1|\theta)}{\int_\theta g_0(\theta)f(z_1|\theta)d\theta}. \tag{9.37}$$

The Bayes estimator of θ is then the expected value of θ based on the posterior distribution (that is, once the information from the observed demand had been incorporated):

$$\theta^* = E[\theta] = \int_\theta \theta g_1(\theta)d\theta. \tag{9.38}$$

The estimator θ^* has several nice theoretical properties. In particular, one can show that it minimizes the variance of the forecast error.

The value θ^* is used in forecasting by setting $\hat{Z}_t = E[Z_t|\theta^*]$. Once the next data value z_2 is observed, we repeat the procedure to get $g_2(\theta)$, and so on. Thus, $g_t(\theta)$ represents our current (time t) belief about θ. (Note that it is a function of the history of observations, $z_1, \ldots, z_t$.)

What makes Bayes estimation practical is that for certain prior distributions of the parameters θ and certain corresponding sample distributions of the random variable *Z,* the posterior distributions of the parameters in (9.37) have the same distributional form as the prior, and their parameters are given by closed-form updating formulas. A pair of distributions that has this property is said to be a *conjugate family of prior distributions.* We list below some well-known pairs of conjugate families of prior distributions (see DeGroot [151] for derivations):

- **Beta-binomial** $Z_1, Z_2, \ldots, Z_N$ are 0-1 random variables from a Bernoulli distribution with $P(Z_t = 1) = \theta$, and θ has a beta distribution with parameters α, β. After observing $z_1, z_2, \ldots, z_N$, θ has a beta distribution with parameters $\alpha + \sum_{k=1}^{N} z_k$ and $\beta + N - \sum_{k=1}^{N} z_k$.

- **Poisson-gamma** $Z_1, Z_2, \ldots, Z_N$ have a Poisson distribution with mean λ, and λ has a Gamma distribution with parameters α, β. After observing $z_1, z_2, \ldots, z_N$, λ has a gamma distribution with parameter $\alpha + \sum_{k=1}^{N} z_k$ and $\beta + N$.

- **Normal-normal** $Z_1, Z_2, \ldots, Z_N$ have a normal distribution with a known variance σ^2 but an unknown mean μ $(\theta = \mu)$, and suppose μ has a normal distribution with mean η and variance v^2. The posterior distribution of μ is a normal distribution with mean

$$\frac{\sigma^2 \eta + v^2 \sum_{k=1}^{N} z_k}{\sigma^2 + N v^2} \tag{9.39}$$

and variance

$$\frac{\sigma^2 v^2}{\sigma^2 + N v^2}. \tag{9.40}$$

The following example illustrates the use of these formulas for forecasting:

Example 9.11 (BAYESIAN FORECASTING) Consider the following time series:

$$Z_t = \mu + \xi_t, \tag{9.41}$$

where ξ_t is normally distributed with a mean of 0 and a known variance σ^2—that is, the random variables $Z_1, Z_2, \ldots$ are assumed to be from a normal distribution $N(\mu, \sigma^2)$.

Suppose our prior distribution on μ is modeled as being normal with mean η_0 and variance $v_0{}^2$. The value η_0 can be thought of as representing our "best guess" of μ and the value v_0 as representing our degree of confidence in this guess.

After an observation z_1 is made, our estimate on the distribution of μ is updated using the update formulas in (9.39) and (9.40).

$$\eta_1 = \frac{\sigma^2 \eta_0 + v_0^2 z_1}{\sigma^2 + v_0^2} \tag{9.42a}$$

$$v_1^2 = \frac{\sigma^2 v_0^2}{\sigma^2 + v_0^2}. \tag{9.42b}$$

After the next observation z_2 is made, they are again updated as follows:

$$\begin{aligned}
\eta_2 &= \frac{\sigma^2\eta_1 + v_1^2 z_2}{\sigma^2 + v_1^2} \\
&= \frac{\sigma^2\eta_0 + v_0^2(z_1 + z_2)}{\sigma^2 + 2v_0^2}, \\
v_2^2 &= \frac{\sigma^2 v_1^2}{\sigma^2 + v_1^2} \\
&= \frac{\sigma^2 v_0^2}{\sigma^2 + 2v_0^2},
\end{aligned}$$

and so forth. After each observation k, the revised forecast of μ is given by

$$E[\mu|z_1, \ldots, z_k] = \eta_k.$$

Notice the ease with which new forecasts can be computed in Example 9.11. The method is also parsimonious with data: only the current estimates need to be stored and updated; all the previous information is contained in the current estimates. However, for distributions that are not conjugate, the updating formulas get complicated, and the Bayesian method loses its attractive properties.

9.3.6.2 Hierarchical and Empirical Bayes Methods

Hierarchical Bayes methods are an appealing way to combine sales data from multiple locations or sources. For example, a manufacturer might be forecasting the sales of its brand across multiple retail chains, a retailer might combine the demand data for a product from multiple stores locations, or an airline might combine data from multiple flights serving a given market.

The method works as follows: Let n be the number of sources and $Z_1, \ldots, Z_n$ represent the random variables of demand at each source. Let $\mathbf{z}_1, \ldots, \mathbf{z}_n$ denote N-vectors of observations of demand at each source ($\mathbf{z}_k$ is assumed to be a vector of N i.i.d. realizations of the random variable Z_k). Let $\theta_1, \ldots, \theta_n$ be the parameters of the distributions of $Z_1, \ldots, Z_n$, respectively, with densities $f_{\theta_k}(\cdot)$. We assume for simplicity that the θ_k's are scalars.

How should we combine these observations? The answer depends on how the parameters θ_k are related. If the parameters $\theta_1, \ldots, \theta_n$ are completely unrelated, we can estimate each independently. If they are all the same, $\theta_1 = \cdots = \theta_n$, we can simply pool all the data together to forecast a single number. However, neither assumption may be satisfactory in a given practical forecasting situation. That is, the sources may be related but not necessarily identical. Hierarchical Bayes methods address this intermediate case. They posit the parameters $\theta_1, \ldots, \theta_n$ as realizations of a common (across the n sources) prior distribution of θ and use the information from "all other" data to obtain a prior for the parameter of each specific source, which is then updated in a Bayesian manner using that source's data.

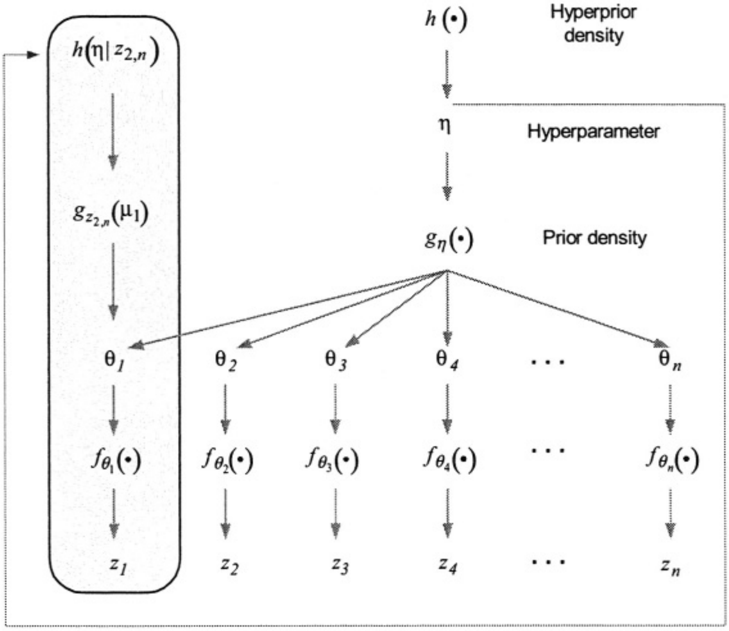


Figure 9.7. The hierarchical Bayes model (adapted from [172]) showing the Bayesian estimation procedure for θ_1 using "other" data, $\mathbf{z}_2, \ldots, \mathbf{z}_n$.

Figure 9.7 shows the hierarchical Bayes model for forecasting θ_1. First, $\theta_1, \ldots, \theta_n$ are assumed to be i.i.d. realizations of a density $g_\eta(\theta)$, where η is a *hyperparameter* from a *hyperprior* density $h(\cdot)$. Both η as well as $h(\cdot)$ are unknown. Then we estimate θ_1 in this framework using not just $\mathbf{z}_1$ but also the other data $\mathbf{z}_{-1} \equiv \{\mathbf{z}_2, \ldots, \mathbf{z}_n\}$. Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ and $\boldsymbol{\theta}_{-1} = (\theta_2, \ldots, \theta_n)$. Using the other data, we perform a Bayesian update on the hyperparameter η to obtain the posterior distribution of

$h(\cdot)$, $h(\eta|\mathbf{z}_{-1})$:[4]

$$h(\eta|\mathbf{z}_{2,n}) \quad \propto \quad h(\eta)\int \prod_{k=2}^{n} g_\eta(\theta_k) \prod_{k=2}^{n} f_{\theta_k}(\mathbf{z}_k)d\boldsymbol{\theta}. \tag{9.43}$$

The integral term in (9.43) is the probability of observing $\mathbf{z}_2, \ldots, \mathbf{z}_n$ for a given η. We can use this to obtain a prior density for θ_1, using Bayes rule:

$$g_{\mathbf{z}_{-1}}(\theta_1) \quad \propto \quad \int h(\eta|\mathbf{z}_{-1})g_\eta(\theta_1)d\eta. \tag{9.44}$$

From this prior density, we calculate the posterior density of θ_1 based on the data set $\mathbf{z}_1$:

$$p_{\mathbf{z}_{2,n}}(\theta_1|\mathbf{z}_1) \quad \propto \quad g_{\mathbf{z}_{-1}}(\theta_1)f_{\theta_1}(\mathbf{z}_1), \tag{9.45a}$$

$$\propto \quad g_{\mathbf{z}_{-1}}(\theta_1)L_1(\theta_1), \tag{9.45b}$$

where $L_1(\theta_1)$ is the likelihood function of θ_1 given $\mathbf{z}_1$. We can interpret $L_1(\theta_1)$ as the information on θ_1 obtained solely from $\mathbf{z}_1$, while $g_{\mathbf{z}_{-1}}(\theta_1)$ is the "correction" based on the information from the other data $\mathbf{z}_{-1}$. Notice that throughout, we do not need to estimate or know the value of η: it is integrated out in (9.44). However, we do need to know the form of the function $h(\cdot)$ to calculate (9.44). This hyperprior density is somewhat removed from the actual data and hence is difficult to interpret or assign a priori.

One way of avoiding specifying the hyperprior density $h(\cdot)$ is to use what is called an *empirical Bayes approximation* to $g_{\mathbf{z}_{-1}}(\theta_1)$. The empirical Bayes approximation proceeds as follows. Suppose we represent the likelihood (with respect to η) given the other data $\mathbf{z}_{-1}$ as

$$L_{\mathbf{z}_{-1}}(\eta) \quad = \quad \int \prod_{k=2}^{n} g_\eta(\theta_k) \prod_{k=2}^{n} f_{\theta_k}(\mathbf{z}_k)d\theta$$

[4] To avoid excessive notation, we do not write down the normalizing factor and just represent the density as being proportional ($\propto$) to the right-hand side. So the use of Bayes rule in (9.43) should be read as

$$h(\eta|\mathbf{z}_{2,n}) \quad = \quad \frac{h(\eta)\int \prod_{k=2}^{n} g_\eta(\theta_k) \prod_{k=2}^{n} f_{\theta_k}(\mathbf{z}_k)d\boldsymbol{\theta}}{\int h(\eta)\int \prod_{k=2}^{n} g_\eta(\theta_k) \prod_{k=2}^{n} f_{\theta_k}(\mathbf{z}_k)d\boldsymbol{\theta}d\eta}$$

$$\propto \quad h(\eta)\int \prod_{k=2}^{n} g_\eta(\theta_k) \prod_{k=2}^{n} f_{\theta_k}(\mathbf{z}_k)d\boldsymbol{\theta}.$$

The density can easily be recovered by dividing by the integral. We do likewise for all subsequent applications of Bayes rule in this section.

and let $\hat{\eta}$ denote the ML estimate

$$\hat{\eta} \;=\; \arg\max_{\eta} \; L_{\mathbf{z}_{-1}}(\eta). \tag{9.46}$$

Then instead of using the exact density $g_{\mathbf{z}_{-1}}(\theta_1)$ of (9.44) in (9.45b), we use the approximation $g_{\hat{\eta}}(\theta_1)$ to obtain the *MLE posterior density* of θ_1

$$p_{\hat{\eta}}(\theta_1|\mathbf{z}_1) \;\;\propto\;\; g_{\hat{\eta}}(\theta_1)L_1(\theta_1).$$

Of course, (9.46) should be easy to solve or else this approximation will be difficult to implement. One can even substitute a MSE estimator or a method-of-moments estimator (Sections 9.2.2 and 9.2.4) instead of the ML estimate of (9.46) if these make the computations more tractable. So one has to choose the densities of $g_{\eta}(\cdot)$ and $f_{\theta}(\cdot)$ judiciously for computational convenience. But in the end, the advantage of this method of empirical approximation is that it does not require an estimate of $h(\cdot)$.

We illustrate the hierarchical Bayes model with a retail RM example:

Example 9.12 (SHRINKAGE ESTIMATION OF RETAIL PRICE AND PROMOTIONAL ELASTICITIES ([75])) A manufacturer sells a product through multiple (n) chains (collection of stores). Periodically the manufacturer offers promotions and wants to gauge the effect of the promotions on sales. The model of sales during a promotional campaign is the following:

$$\begin{aligned} SL_t = \theta_1 + \theta_2 PR_t + \theta_3 DD_{t-1} + \theta_5 AD_t \\ + \theta_6 DP_t + \theta_7 FL_t + \theta_8 CD_t + \theta_9 SL_{t-1} + \xi_t, \end{aligned} \tag{9.47}$$

where

SL_t = logarithm of sales in period t

PR_t = relative price in period t (regular price divided by an average of competitive regular prices)

DD_t = deal discount in period t (normal shelf price minus actual divided by normal shelf price)

AD_t = feature advertising in period t (proportion of stores in chain using the ad)

DP_t = display in period t (proportion of stores in chain displaying the brand)

FL_t = 0-1 indicator variable, 1 if period t is the final period of a multi-week deal, and 0 otherwise

CD_t = maximum deal discount for competing brands in chain in period t

The data consists of T periods of sales data from the n chains. Let z_t^i represent the log sales of chain-brand i at time t, and y_{mt}^i, the m^{th} covariate (explanatory variable in (9.47), $m = 1,\ldots,M$ $(M = 7)$) value for period t for chain i. The regression models for the log sales for the n chains are given by

$$Z_t^i \;=\; \sum_{m=1}^{M} \theta_m^i y_{tm}^i + \xi_t^i, \quad i = 1,\ldots,n; \;\; t = 1,\ldots,T,$$

where the ξ_t^i are assumed to have a normal distribution with mean zero and a common variance σ^2, and to be independent. Let $\boldsymbol{\theta}^i = (\theta_1^i, \ldots, \theta_M^i)$, and $\boldsymbol{\xi}^i = (\xi_1^i, \ldots, \xi_{p_i}^i)$. We assume σ^2 is known,[5] and let $\boldsymbol{\sigma}^2$ denote $\sigma^2\mathbf{I}$.

This is a straightforward regression problem if the n stores are estimated separately. Let $\mathbf{Z}^i = (Z_1^i, \ldots, Z_T^i)$, and $\mathbf{Y}^i$ the matrix whose elements are y_{tm}^i, $t = 1, \ldots, T$; $m = 1, \ldots, M$. Then the regression equation for store i in matrix form is

$$\mathbf{Z}^i = \mathbf{Y}^i\boldsymbol{\theta}^i + \boldsymbol{\xi}^i.$$

The MSE estimates for $\boldsymbol{\theta}^i$ are given by (same as (9.6))

$$\hat{\boldsymbol{\theta}}^i = ((\mathbf{Y}^i)^\mathsf{T}\mathbf{Y}^i)^{-1}(\mathbf{Y}^i)^\mathsf{T}\mathbf{Z}^i.$$

However, estimating by chain reduces the size of the data sets and often leads to odd predictions with wrong signs on the coefficients or similar calibration problems.

We can build instead a hierarchical model assuming that each parameter θ_m^i comes from a prior normal distribution

$$\theta_m^i \sim N(\mu_m, \varsigma_m), \quad i = 1, \ldots, n; \ m = 1, \ldots, M.$$

$\eta_m = \{\mu_m, \varsigma_m\}$, $m = 1, \ldots, M$, are the hyperparameters with an unknown distribution $h(\cdot)$, generating the parameters θ_m^i, $m = 1, \ldots, M$; $i = 1, \ldots, n$. Let $\boldsymbol{\Sigma} = \mathrm{diag}[\varsigma_1, \ldots, \varsigma_M]$, and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_M)$.

If we knew $\eta_m = \{\mu_m, \varsigma_m\}$, $m = 1, \ldots, M$, *and* we had a prior $\hat{\boldsymbol{\theta}}^i$ of the parameters, then we could have estimated the mean of the posterior distribution of $\boldsymbol{\theta}^i$ by Bayes theorem as (using a vector version of the formulas in Example 9.11)

$$\mathbf{Q}_i^{-1}(\sigma^{-2}(\mathbf{Y}^i)^\mathsf{T}\mathbf{Y}^i\hat{\boldsymbol{\theta}}^i + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}), \tag{9.48}$$

where

$$\mathbf{Q}_i = \sigma^{-2}(\mathbf{Y}^i)^\mathsf{T}\mathbf{Y}^i + \boldsymbol{\Sigma}^{-1}.$$

The new updated mean of (9.48) is in a sense a convex combination of the prior mean $\hat{\boldsymbol{\theta}}$ and the actual (unknown) mean $\boldsymbol{\mu}$. The mean is "shrunk" toward the hyperparameter $\boldsymbol{\mu}$ by the *shrinkage factor* $\mathbf{Q}_i^{-1}\sigma^{-2}(\mathbf{Y}^i)^\mathsf{T}\mathbf{Y}^i$.

The estimate (9.48) is unusable however as we do not know $\eta_m = \{\mu_m, \varsigma_m\}$, $m = 1, \ldots, M$ (i.e., $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$). If we had estimates of the hyperparameters from "other" data however, we can use them instead in (9.48) for any given chain i. So the estimates of $\boldsymbol{\theta}^i$ would be a convex combination of a hyperprior estimate of μ's from data other than from chain i and the data of chain i. This is the idea behind the hierarchical Bayes method.

In practice, obtaining the ML estimates of η from the other data may be too difficult. But this does not prevent us from using any reasonable estimate that we can obtain based on the other data. Blattberg and George [75] give a variety of alternatives for the hyperparameter estimates for this regression problem. Also, here

[5] For any given set of estimators of $\boldsymbol{\theta}^i$, $\hat{\boldsymbol{\theta}}^i$, a good estimate of σ^2 is

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n \sum_{t=1}^T (z_t^i - \sum_{m=1}^M \hat{\theta}_m^i y_{tm}^i)^2}{\sum_{i=1}^n T - nM + 2}.$$

we have constrained the θ_m^i's to have identical variances ς_m. See [75] for alternative constraints with different interpretations. Blattberg and George [75] also consider weekly sales data for a national brand and show that hierarchical Bayes methods improve predictive performance.

9.3.7 State-Space Models and Kalman Filtering

Like time-series methods, state-space methods assume the time series $\{Z_t\}$ is driven by an underlying dynamic system. The system is defined by a "state" together with a system of equations for describing how the state and observable outputs (say, the time-series data) evolve over time as function of possibly random inputs. The future behavior of the system can be completely described by the present state and future inputs, a feature known as a *Markovian representation* of the system. However, the current state most often is not directly observable and must be estimated based on observed data. The following example illustrates a simple case of such a system:

Example 9.13 Consider a series being generated by the following model:

$$Z_t = \mu_t + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2) \tag{9.49a}$$

$$\mu_t = \phi\mu_{t-1} + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2), \tag{9.49b}$$

where μ_t is the underlying mean of data Z_t. Here the mean μ_t (a scalar) is the state of the system, which we cannot observe directly. The mean evolves according to the state equation (9.49b), which is a linear function of the past state (μ_{t-1}) and a process noise term η_t. The observable output Z_t is described by the observation equation (9.49b) and is equal to the mean μ_t plus a measurement noise term ξ_t.

For a time series generated by (9.49a)–(9.49b), a forecasting method might proceed as follows: (i) keep a current estimate of the underlying state $\hat{\mu}_t$, (ii) forecast $\hat{Z}_{t+1} = \hat{\mu}_t$, (iii) after observing the data at time $t + 1$, update our current estimate of state to $\hat{\mu}_{t+1}$ and repeat. (Details of how this can be done are discussed below.)

One can view many forecasting models in a state-space framework. For example, in simple exponential smoothing equation (9.19), the level factor A_t can be interpreted as the unobservable state, while Bayesian forecasting methods can be viewed as an attempt to estimate an unobservable "state" (the unknown parameters of the distribution). More generally, if we define the "state" at time t as consisting of the complete history of observations and actions up to time t, then this state would contain all the information relevant for forecasting. Thus, at an abstract level, all forecasting models can be cast in a state-space model framework. However, such an abstract description is of little practical value because the dimension of the state increases without bound over time. Hence, for the state-space approach to be useful, we need a more compact (finite-dimensional) representation of the state, as in

Example 9.13. In this section, we focus on the best known state-space forecasting method: the Kalman filter.

9.3.7.1 The Kalman Filter Formulation

The Kalman filter is based on a finite-dimensional system of linear state and observation equations and zero-centered Gaussian (normally distributed) noise terms. Under these conditions, the Kalman filter provides an efficient algorithm for estimating the state and for forecasting.

Formally, let the n dimensional real vector $\mathbf{y}_t$ represent the state at time t. The state is assumed to evolve according to a linear system equation:

$$\mathbf{y}_t = \mathbf{A}\mathbf{y}_{t-1} + \boldsymbol{\nu}_t, \tag{9.50}$$

where $\boldsymbol{\nu}_t$ is a n-vector of random variables, called the *process noise,* and $\mathbf{A}$ is a known $n \times n$ matrix of parameters. We assume $\boldsymbol{\nu}_t$ is a Gaussian *(white-noise)* process—a set of i.i.d. random variables from a normal distribution $N(\mathbf{0}, \mathbf{Q})$, where $\mathbf{Q}$ is a known $n \times n$ matrix called the *process-noise covariance* matrix.

There is a m-dimensional vector $\mathbf{z}_t$ of observations,[6] which is related to the state by the following observation equation:

$$\mathbf{z}_t = \mathbf{H}\mathbf{y}_t + \boldsymbol{\xi}_t,$$

where $\mathbf{H}$ is a known $m \times n$ matrix of parameters, and $\boldsymbol{\xi}_t$ is a m-vector of i.i.d. random variables, called the *measurement noise,* that we assume has a normal distribution $N(0, \mathbf{R})$, with a known $m \times m$ *measurement noise covariance* matrix $\mathbf{R}$. While we assume the matrices $\mathbf{A}, \mathbf{H}, \mathbf{Q},$ and $\mathbf{R}$ are known, in practice they are usually estimated from data as discussed later.[7] To illustrate this formulation, we give an example of the AR(2) model in state-space form:

Example 9.14 Consider the AR(2) process described in Section 9.3.2, where

$$\mathbf{z}_t = \delta + \xi_t + \theta_1 z_{t-1} + \theta_2 z_{t-2}. \tag{9.51}$$

[6] Note that the observation is a vector here, in contrast to the scalar observations of previous sections. We also use **z** and **y** to represent the *random variables* generating **z** and **y**, instead of **Z** and **Y** as in the rest of this chapter, to avoid confusion with our matrix notation convention.

[7] Here we have also assumed that the matrices **A**, **H**, **Q**, **R** are constant across time. However, the theory and the Kalman filter forecasting equations hold even when this data changes over time. The Gaussian distribution assumption on the error terms is also not strictly necessary, although it is commonly assumed in most applications.

We can rewrite equation (9.51) as a system of state-space equations, as a combination of a state-evolution equation,

$$\overbrace{\begin{bmatrix} z_t \\ z_{t-1} \\ \delta \end{bmatrix}}^{\mathbf{y}_t} = \overbrace{\begin{bmatrix} \theta_1 & \theta_2 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{\mathbf{A}} \overbrace{\begin{bmatrix} z_{t-1} \\ z_{t-2} \\ \delta \end{bmatrix}}^{\mathbf{y}_{t-1}} + \overbrace{\begin{bmatrix} \xi_t \\ 0 \\ 0 \end{bmatrix}}^{\boldsymbol{\nu}_t},$$

and as a measurement equation,

$$z_t = \overbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}^{\mathbf{H}} \overbrace{\begin{bmatrix} z_t \\ z_{t-1} \\ \delta \end{bmatrix}}^{\mathbf{y}_t} + \overbrace{0}^{\xi_t}.$$

In a similar fashion, the general $\mathrm{ARMA}(p,q)$ model can also be formulated in a Kalman-filter framework (see Wei [560], p.385), as can many of the other time-series models of Section 9.3.2 (see Harvey [243]).

In a forecasting context, the state can be viewed as the (unobservable) parameters of the true underlying demand-generation process. Each observation gives additional information of the parameters, and this information can be used to update our current estimate of the state via the state-evolution equation. With the updated state, a forecast for period $t+1$ can be made using the prediction equation for period $t+1$, substituting the state obtained for period t. The Kalman filter provides an efficient recursive algorithm for performing these operations.

9.3.7.2 The Kalman Filter Forecasting Algorithm

We first state the Kalman filter forecasting algorithm, and then explain the intuition behind it and some of its formal properties.

The algorithm proceeds as follows. Let the subscript indexing $(\cdot)_{t|t-1}$ denote the value of the variable at time t based on all the information up to time $t-1$ (before the observation in period t). At each time t, we keep an estimate of the underlying state $\hat{\mathbf{y}}_{t|t-1}$ that encapsulates all the information gained from past observations. After time t, we get a new observation $\mathbf{z}_t$ and update our estimate of state to $\hat{\mathbf{y}}_{t|t}$ using $\hat{\mathbf{y}}_{t|t-1}$ and $\mathbf{z}_t$ (by (9.50)). We then make a forecast for time $t+1$, $\hat{\mathbf{z}}_{t+1} = \mathbf{H}\hat{\mathbf{y}}_{t+1|t}$, with $\hat{\mathbf{y}}_{t+1|t} = \mathbf{A}\hat{\mathbf{y}}_{t|t}$.

Let $\mathbf{e}_{t|t-1} = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}$ and $\mathbf{e}_{t|t} = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t}$ represent, respectively, errors from the true state before and after the state estimates have been updated. Let $\mathbf{P}_{t|t-1} = E[\mathbf{e}_{t|t-1}\mathbf{e}_{t|t-1}^{\mathsf{T}}]$ and $\mathbf{P}_{t|t} = E[\mathbf{e}_{t|t}\mathbf{e}_{t|t}^{\mathsf{T}}]$ represent, respectively, the error covariance matrices. The algorithm is as follows:

Initialization: Let time $t = 0$. Assume initial values of $\mathbf{P}_{0|0}$ (say $\mathbf{I}$) and the initial state $\hat{\mathbf{y}}_{0|0}$.

Forecasting step: At time t, project the error, state, and forecast:

$$\begin{aligned}
\hat{\mathbf{y}}_{t+1|t} &= \mathbf{A}\hat{\mathbf{y}}_{t|t} \\
\mathbf{P}_{t+1|t} &= \mathbf{A}\mathbf{P}_{t|t}\mathbf{A}^{\mathsf{T}} + \mathbf{Q} \\
\hat{\mathbf{z}}_{t+1} &= \mathbf{H}\hat{\mathbf{y}}_{t+1|t}.
\end{aligned}$$

Measurement updating step: After observing $\mathbf{z}_{t+1}$, update

$$\hat{\mathbf{y}}_{t+1|t+1} = \hat{\mathbf{y}}_{t+1|t} + \mathbf{K}_{t+1}(\mathbf{z}_{t+1} - \hat{\mathbf{z}}_{t+1}), \tag{9.52}$$

where the matrix $\mathbf{K}_{t+1}$ is given by

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{H}^{\mathsf{T}}(\mathbf{H}\mathbf{P}_{t+1|t}\mathbf{H}^{\mathsf{T}} + \mathbf{R})^{-1}.$$

Update the error covariance

$$\mathbf{P}_{t+1|t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H})\mathbf{P}_{t+1|t}.$$

The matrix $\mathbf{K}_t$ is known as the *Kalman gain.* The crucial step is (9.52), which calculates the a *posteriori* estimate of the state *after* observing the measurement in period $t+1$ from the *a priori* estimate (*before* observing the measurement in period $t+1$). If the disturbances are normal, the distribution of the initial state will be normal, and the mean and variance of the a *priori* estimate of the state are given by $\hat{\mathbf{y}}_{t+1|t}$ and $\mathbf{P}_{t+1|t}$. The conjugate distribution of a normal distribution is again normal and after observing the measurement $\mathbf{z}_{t+1}$, the *a posteriori* distribution of $\mathbf{y}_{t+1|t+1}|\mathbf{z}_{t+1}$ is also normal with mean given by (9.52). This mean-state vector also turns out to be the minimum mean-square estimate of $\mathbf{y}_{t+1|t+1}$ given all the information up to time $t+1$. Even when the disturbances are not normal, the Kalman filter equations can be shown to be the best *linear* estimator, in the sense of minimizing the mean-square error among all linear updates of the form $\hat{\mathbf{y}}_{t+1|t+1} = \hat{\mathbf{y}}_{t+1|t} + \mathbf{K}(\mathbf{z}_{t+1} - \hat{\mathbf{z}}_{t+1})$; that is, the Kalman gain is the matrix $\mathbf{K}$ that minimizes $(\mathbf{z}_{t+1} - \mathbf{H}\hat{\mathbf{y}}_{t+1|t+1})^{\mathsf{T}}(\mathbf{z}_{t+1} - \mathbf{H}\hat{\mathbf{y}}_{t+1|t+1})$.

An attractive property of the Kalman filter is the recursive nature of the algorithm. At each step, we need only to maintain the current estimate of the state and the estimate of the covariance matrix. As new observations come in, we can then easily update these two quantities.

Moreover, updating these estimates by the Kalman filter equations is computationally very efficient, which is one of the most appealing features of the algorithm. The following is a simple example of the operation of the Kalman filter:

Example 9.15 (Forecasting using the Kalman filter) Let the state evolution equations for a 1-dimensional state be given by

$$y_t = y_{t-1} + \nu_t$$

and the measurement be given by the process

$$z_t = y_t + \xi_t,$$

where $\nu_t \sim N(0, Q)$ and $\xi_t \sim N(0, R)$. Then the state update equations of the Kalman Filter are

$$\begin{aligned} \hat{y}_{t+1|t} &= \hat{y}_{t|t} \\ P_{t+1|t} &= P_{t|t} + Q \\ \hat{z}_{t+1} &= \hat{y}_{t+1|t}, \end{aligned}$$

and the measurement equations to update the state and measurement are

$$\hat{y}_{t+1|t+1} = \hat{y}_{t+1|t} + K_{t+1}(z_{t+1} - \hat{z}_{t+1}), \tag{9.53}$$

where K_{t+1} is given by

$$K_{t+1} = \frac{P_{t+1|t}}{(P_{t+1|t} + R)}.$$

Update the error covariance by

$$P_{t+1|t+1} = (1 - K_{t+1})P_{t+1|t}.$$

To start off the forecasting process, at $t = 0$, we need to assign some values to $\hat{y}_{0|0}$ and $P_{0|0}$. Rather arbitrarily let's set $\hat{y}_{0|0} = 1$. As with Bayesian methods, the quantity $P_{0|0}$ should reflect our degree of certainty about our estimate of the state $\hat{y}_{0|0}$. A value of $P_{0|0} = 0$ would imply that we are completely sure of our initial estimate; more often, we choose some value $P_{0|0} \neq 0$. The precise value is not critical—the Kalman filter algorithm is quite robust this way—but the more uncertain we are of our estimate, the higher this value should be (something like $P_{0|0} = 2$ would generally suffice for this case).

Notice the similarity between (9.53), which can be rewritten in terms of z and $\hat{z}$'s as

$$\hat{z}_{t+1} = K_t z_t + (1 - K_t)\hat{z}_t$$

and the simple exponential smoothing formula (9.19), repeated here:

$$\hat{z}_{t+1} = \alpha z_t + (1 - \alpha)\hat{z}_t.$$

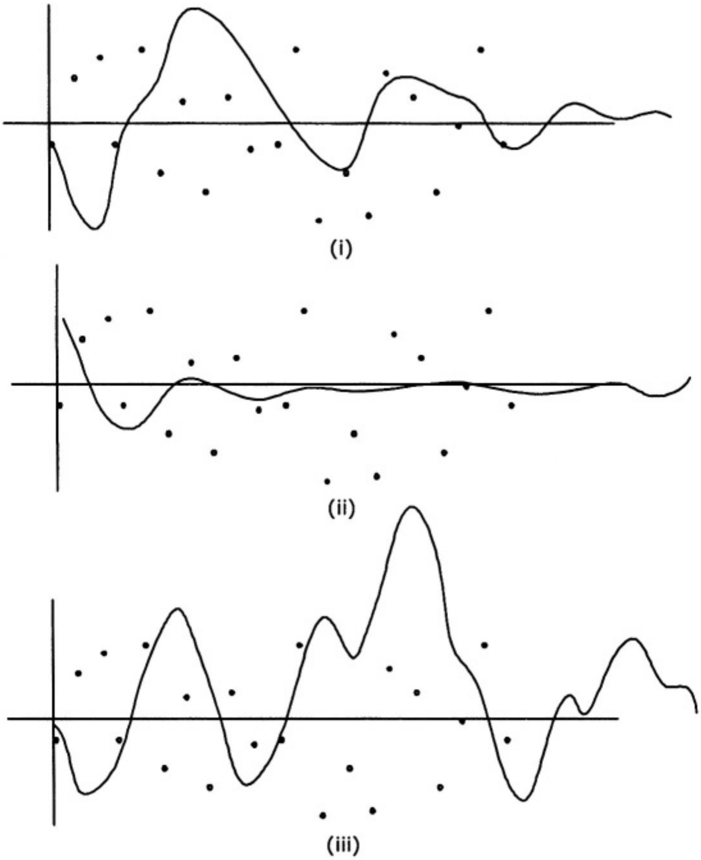


Figure 9.8. Kalman filter smoothing. (i)A data stream generated with a measurement error ξ around the mean and the Kalman filter uses a value $R = Var(\xi)$ (ii) Kalman filter uses a value of $R > Var(\xi)$ (the filter is smoothed too much), (iii) the Kalman filter uses a value $R < Var(\xi)$, and the forecasts follow the noise too quickly.

Indeed, the Kalman gain K_t can be considered as an adaptive smoothing factor that changes over time based on the observed data. As $t \rightarrow \infty$, one can also show the Kalman gain converges to a constant matrix *K,* which means, after many observations, the Kalman filter will converge to the simple exponential smoothing formula (9.19). However, the Kalman gains are in fact the “optimal” weighting factors, in the sense that for linear state and measurement processes, they minimize the mean-square error.

9.3.7.3 Estimating the Matrices A, H, Q, and R

Lastly, we address the question of estimating the matrices **A, H, Q,** and **R.** Although the Kalman filter equations are easy to apply if these matrices are known, in practice it is highly unlikely that we know their exact values. For instance, in the state-space formulation of the AR(2) process (9.51), we do not know the components θ_1 and θ_2 of the matrix **A.** However, these parameters can be estimated by maximum-likelihood methods based on an initial set of observations (see Harvey [242] and Harvey [243], p.91). The values used for **Q** and **R** will also affect the behavior of the algorithm. If the values we choose for **Q** and **R** are much higher than the true variance in the process and measurement error terms, then the forecasts tend to be very reactive to noise, and if they are much smaller than the actual variances, the forecasts are much smoother (see Figure 9.8). Again, these variances can also be estimated by maximum-likelihood methods.

9.3.8 Machine-Learning (Neural-Network) Methods

All the forecasting methods we have discussed thus far follow the same underlying strategy: posit a functional form for the relationship between the observed data and various factors (such as noise terms, time, past observations, and causal factors) and then estimate the parameters of this function using historical data. In contrast, *machine-learning*—or specifically, *neural-network*—methods do not make a functional assumption *a priori;* rather, they use interactions in a network-processing architecture to automatically identify the underlying function that best describes the demand process. The methods are based on artificial intelligence approaches that mimic the way the human brain learns from experience. In theory, with the appropriate architecture and training procedure, neural networks are capable of approximating any nonlinear functional form after a sufficient degree of “learning” on samples generated by that function.

Neural networks have found wide applicability in pattern recognition, classification, reconstruction, biology, computer game playing, and time series forecasting. Business applications have been reported in market analysis, bond rating, credit-risk evaluation, and financial series forecasting. Some RM vendors and airlines have implemented neural-network forecasting methods as well [496].

Neural-network forecasting encompasses a large class of architectures and algorithms, and the literature is extensive. Here we only describe the workings of a simple neural network with the most basic of training

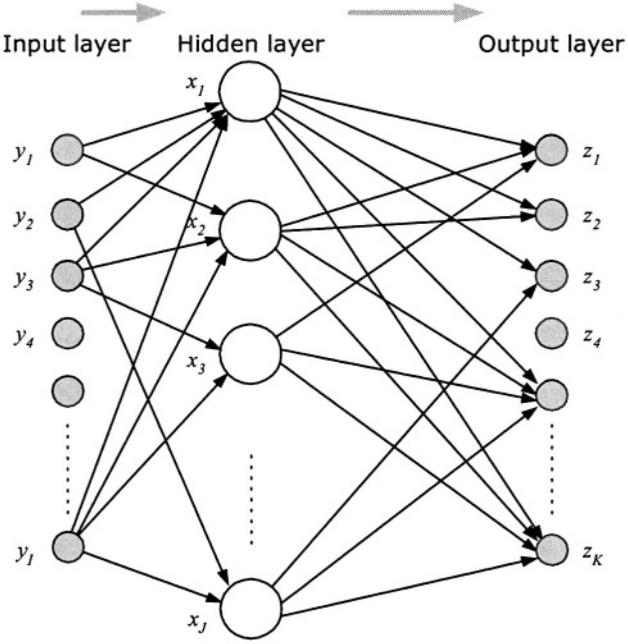


Figure 9.9. A three-layer feed-forward neural network.

algorithms. However, this introduction should provide a good sense of the overall approach.

9.3.8.1 An Overview of Neural Networks

A neural network consists of an underlying directed graph and a set of additional quantities defined on the graph. In an important class of neural networks, the nodes of the network are arranged in consecutive layers, and the arcs are directed from one layer to the next, left to right as shown in Figure 9.9. Such networks are called *feed-forward* networks or *perceptrons* and form the most important class of neural networks used for forecasting. We limit our discussion here to feed-forward networks.

The first layer is called the *input layer* and the last is called the *output layer,* with the layers in the middle being the *hidden layers.* Most networks in practice have at most one or two hidden layers. A network with a single hidden layer has been shown to be able to approximate most nonlinear functional forms [397]. The training data is "fed" to the input layer, and the forecasts are "read" from the output layer. Typically, in demand-forecasting applications, each node in the input layer corresponds to an explanatory variables (analogous to the y's in the linear-regression equation (9.3)), and each node in the output layer corresponds to a future forecast. For example, if we want to use the 20 most recent historical observations to make forecasts for the next three periods, the network would have 20 input nodes (one for each historical observation) and three output nodes (one for each forecast), with a certain number of hidden nodes in between.

More generally, a neural-network architecture is defined by a graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is a set of nodes and $\mathcal{A}$ is a set of directed arcs. The following quantities are defined on the network:

- A *state variable,* s_j, associated with each node $j \in \mathcal{N}$. Typically, state is binary (every node is either active (state 1) or inactive (state 0)) or it is continuous, usually taking on values between 0 and 1. The state can change for each set of inputs or in an online forecasting application after every new observation. Thus, states are said to *evolve* over discrete units of time t, $t = 0, 1, 2, \ldots$, and we represent the state of node j at time (observation) t as $s_j(t)$.

- A *weight,* w_{ij}, associated with each directed arc $(i, j) \in \mathcal{A}$.

- An *activation threshold value* ν_j associated with each node $j \in \mathcal{N}$. Typically, the activation threshold value serves as a threshold for making the node active or inactive. For example, if the sum of the weights of incoming arcs exceeds ν_j, then consider node j active and inactive otherwise.

- An *activation function* (or *transfer function),* which determines the state of node j as a function of the states of other nodes i with arcs into j (with arcs of the form (i, j)), the arc weights w_{ij}, and the activation threshold ν_j: $f_j(\{s_i, w_{ij} : (i, j) \in \mathcal{A}\}, \nu_j)$. The activation functions can be different for each layer (or even each node). Typically, the activation functions act on the *sum* of the weights of arcs from *active* nodes coming into node j, in which case, the activation threshold for j can be represented as $f_j(\sum_{\{(i,j) \in \mathcal{A}} w_{ij} s_i - \nu_j)$. Activation functions serve to make the nodes active or inactive.

  Some examples of transfer functions f include the following:

  - A *linear function,* where $f(h) = h$.
  - The *Heavyside step function,* which is a simple threshold value comparison between $\sum_{\{(i,j) \in \mathcal{A}} w_{ij} s_i$ and ν_j:

$$f(h) = \begin{cases} 1 \text{ (active)} & \text{if } h \geq 0 \\ 0 \text{ (inactive)} & \text{otherwise.} \end{cases}$$

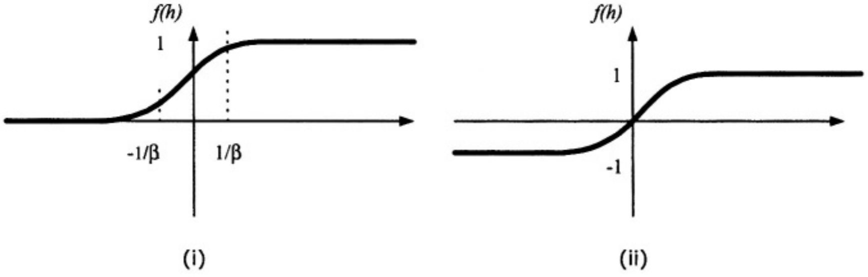


Figure 9.10. The (i) Fermi function and the (ii) tanh activation functions.

- The *logistic sigmoid functions* (Figure 9.10), which are a class of monotonic, differentiable functions $f(h)$ with

$$\lim_{h \to -\infty} f(h) = 0,$$

and

$$\lim_{h \to \infty} f(h) = 1.$$

An example of a logistic sigmoid function is the *Fermi function:*

$$f(h) \quad = \quad \frac{1}{1+e^{-2\beta h}}. \tag{9.54}$$

- The tanh function:[8]

$$f(h) \quad = \quad \tanh(\beta h) = \frac{1-e^{-2\beta h}}{1+e^{-2\beta h}}. \tag{9.55}$$

The value of the transfer function is taken to be the *state* of the node. The state is binary (0 or 1) for the Heavyside step function and continuous for the linear function (from $-\infty$ to ∞) and the logistic sigmoid functions (between 0 and 1).

9.3.8.2 Training and Forecasting

Calibration of a neural network is called *training the network.* A set of training data is used to calibrate the weights and the values of the threshold functions. Once these parameters are determined, the network can be used for forecasting. Thus, the three main steps are defining the

[8] The tanh function can be shown to be equivalent to the Fermi function after a linear transformation of the inputs and outputs (see Bishop [69], p.127). However, the tanh function has been found to give faster training convergence and is generally preferred.

network, training, and forecasting. We illustrate these steps on the three-layer network of Figure 9.9.

Defining the Network The input is a set of I values of independent variables associated with each observation, represented by I input nodes, and the output is a forecast for K future periods, represented by K output nodes. The inputs could consist of all variables that would influence the demand. For instance, if the forecast is for demand in a particular market for an airline, the input variables, in addition to historical demand in that market, could consist of variables such as schedule frequency, capacity, time in market or economic indicators. Assume there are J nodes in the hidden layer. We index arcs from the input layer to the hidden layer as (i, j) and arcs from the hidden layer to the output layer as (j, k).

We next need to define the transfer functions. We use the tanh function (9.55) as the activation function $f(h)$ for the nodes of the hidden layer and a linear function $\tilde{f}(h) = h$ as the activation function for the nodes of the input and output layers. These functions are defined by the activation thresholds ν_i, the arc weights w_{ij}, w_{jk}, and the parameter β of the tanh function.

Let y_i represent the state of input node i, and x_j the state of node j of the hidden layer, and $\hat{Z}_k$ the state of node k of the output layer. The inputs to the hidden layer are formed by a weighted combination of values of the states of the input layer

$$h_j = (\sum_{i=1}^{I} w_{ij} y_i) - \nu_j,$$

and the state of the hidden node j is therefore

$$x_j = f(h_j).$$

The inputs to the output layer in turn are a weighted combination of the states of the hidden layer and the activation thresholds of the output nodes:

$$h_k = (\sum_{j=1}^{J} w_{jk} x_j) - \nu_k.$$

The state of the output node k is then $\tilde{f}(h_k) = h_k$. This completes the definition of the network.

Training Once a network topology is chosen, we have to determine values for the arc weights and node activation thresholds. This training is, in all respects but terminology, equivalent to estimating the parameters of any other forecasting model from historical data—except that we are not working with a simple functional form for the demand generating process but rather from a complicated network of interacting functions.

One of the first, and still quite popular, methods of training is the *error back-propagation method.* The method uses a squared error criterion and prescribes an iterative procedure to update the weights to minimize the squared error. Appendix 9.A gives an application of this algorithm to the three-layer network of Figure 9.9.

Forecasting Once training is complete, we have a set of values for the parameters of the network, ν_i, w_{ij}, w_{jk}, and the parameters β of the tanh function. Since the state of the input nodes y_i is equal to $\tilde{f}_i(h)$, and since we chose $\tilde{f}$ to be the linear function, the input state is simply the input to node m. Again, the inputs to the hidden layer are a weighted combination of values of the states of the input layer

$$h_j = (\sum_{i=1}^{I} w_{ij} y_i) - \nu_j,$$

so the state of the hidden node j is computed as

$$x_j = f(h_j).$$

The inputs to the output layer are again a weighted combination of the states of the hidden layer and the activation thresholds of the output nodes:

$$h_k = \sum_{j=1}^{J} w_{jk} x_j - \nu_k.$$

The final forecast is then given by state of the output nodes:

$$\hat{Z}_k = \tilde{f}(h_k) = h_k.$$

9.3.8.3 More Advanced Neural Networks

The network architecture and training algorithms described thus far form the most basic neural-network methodology. But other variations of this method are available. Even for the simple method presented here, we have not delved into procedures to choose the number of hidden layers or the number of nodes in each layer, or the best choice of the transfer functions. For example, there are many procedures to automatically prune or grow the network topology based on the observed data and the network's predictive performance.

As far as training goes, we have described only one of the earliest and the most basic of training algorithms. A significant amount of the neural network literature is devoted to improving training, in terms of speeding up the convergence or ensuring the convergence is to the right parameters (global convergence), and avoiding overfitting (Section 9.5.1.4). The interested reader should consult a textbook on neural networks before deciding among these various options.

9.3.9 Pick-up Forecasting Methods

Pick-up forecasting methods exploit some unique characteristics of reservation data in quantity-based RM, where the period between repeated service offerings is shorter than the period over which reservations are made (for example, an airline offers a daily flight between two cities but accepts reservations for these flights up to 90 days prior to departure). They are best viewed as a forecasting strategy—specifying a method for disaggregating and aggregating reservations data—rather than a class of fundamentally new forecasting algorithms.

As we mentioned in Section 9.1.3.4, reservations data has a "wedge-shaped" form, in which one has a partial and evolving picture of demand over time. Figure 9.11 shows this evolution of demand in matrix and graphical form for resources sold on consecutive dates. Rather than relying only on complete booking histories for forecasting, pick-up methods exploit both the complete and partial-bookings data to make better forecasts. The main idea is to forecast incremental bookings (booking obtained over short intervals of time prior to service) and then aggregate these increments to obtain a forecast of total demand to come.

We illustrate this idea with the *additive pick-up method.* Suppose for the data in Figure 9.11, we want to forecast for 13-June when we have one day remaining. The historical observed bookings on the day of departure are 8, 2, and 13 (for the service dates 12 June, 11 June and 10 June respectively). From this data {8, 2, 13}, we make an incremental forecast for zero-day prior for 13 June (bookings expected on 13 June) as, say, the mean value of 7.6. Similarly, for the forecast for 14 June, we first construct two incremental forecasts, one for zero-day prior and the other for one-day prior; sticking to our averaging method, this yields incremental forecasts of 7.6 and 3.75, respectively. Then the forecast of demand to come for 14 June is the sum of these two increments or 7.6 + 3.75 = 11.35, and so on, for the other dates in the future.

Formally, the k-day ahead forecast of demand to come is given by

$$\hat{Z}(t+k) = \sum_{i=0}^{k} \hat{Z}_{[i]}(t+k),$$

Number of Days Prior to the Usage of the Resource									Resource-Usage Date
-8	-7	-6	-5	-4	-3	-2	-1	0	
6	3	11	4	9	8	13	3	13	10-Jun
8	6	6	3	16	11	5	4	2	11-Jun
1	2	0	0	3	6	2	6	8	12-Jun
6	0	4	1	2	6	3	2		13-Jun
3	8	8	6	5	1	2			14-Jun
1	0	2	7	6	4				15-Jun
0	1	1	6	5					16-Jun
1	11	12	6						17-Jun

Incremental bookings

Number of Days Prior to the Usage of the Resource									Resource-Usage Date
-8	-7	-6	-5	-4	-3	-2	-1	0	
6	9	20	24	33	41	54	57	70	10-Jun
8	14	20	23	39	50	55	59	61	11-Jun
1	3	3	3	6	12	14	20	28	12-Jun
6	6	10	11	13	19	22	24		13-Jun
3	11	19	25	30	31	33			14-Jun
1	1	3	10	16	20				15-Jun
0	1	2	8	13					16-Jun
1	12	24	30						17-Jun

Cumulative bookings

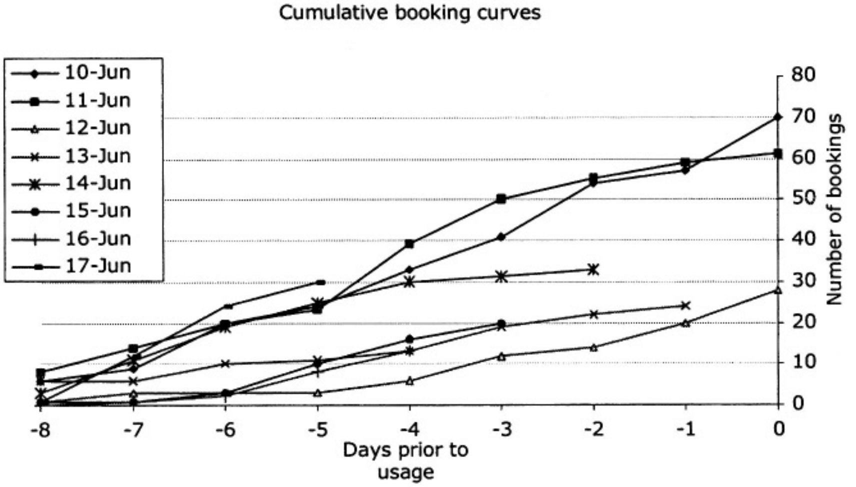


Figure 9.11. **Incremental bookings, cumulative bookings, and cumulative booking curves for eight consecutive days of a resource. 12 June is the current date with full historical bookings. 13 June till 17 June have partial booking information.**

where $\hat{Z}_{[i]}(\cdot)$ represents the incremental bookings forecast i days prior to the time of service. The forecasts $\hat{Z}_{[i]}(t+k)$ are constructed using the available historical i-**day-prior** incremental bookings. In principle, any time-series method can be used to make these incremental forecasts.

In the *multiplicative pick-up method,* the forecast is performed on data normalized as a fraction of current bookings. So if k days prior to the resource usage date there are 100 total bookings on hand and on $(k-1)$ days prior 10 bookings were observed, then the incremental increase is 10% or 0.1. The incremental bookings data is first converted to such fractions. In our example in Figure 9.11, to make a forecast for 13 June, we convert the zero-day prior incremental bookings into $\{8/14, 2/55, 13/54\}$ (14, 55, and 54 are the total bookings on hand for 12 June, 11 June, and 10 June, respectively). Similarly, the one-day prior fractions for 14 June are $\{2/22, 6/14, 4/55, 3/54\}$. We can take the average of these fractions to obtain the forecast of the pick-up fraction zero-day prior and one-day prior. This would be 0.284 for zero-day prior and 0.162 for one-day prior, the average multiplicative "pick-up" over current bookings. A forecast of demand to come for 14 June would be $0.284 \times (33 + 0.162 \times 33) + 0.162 \times 33 = 16.23$. This is higher than given by the additive pick-up method, reflecting the underlying assumption of the multiplicative method that future bookings are positively correlated with current bookings. Other aggregation strategies and variations are possible.

Again, the advantage of pick-up methods is that they use all the available bookings information. Moreover, as partial bookings are recent data, using this data can make the forecast more responsive to shifts in demand. While the idea is simple and mostly heuristic, pick-up methods are widely used in quantity-based RM and reported to perform well.

9.3.10 Other Methods

Several other methods of forecasting have been reported in RM. The *Delphi method* is a formal procedure for extracting analyst and managers' opinion on expected demand. It is used primarily in cases where there is no historical information, where there is an unexpected demand shock, or in some cases when RM is done manually. Fuzzy logic (Ting and Tzeng [512]) and expert systems (Basgall [29]) have been proposed as the basis for a second level of automation in RM forecasting. These systems attempt to replicate the rules used by human analysts when monitoring and overriding a RM system. Chaos-theoretical models for forecasting market response have been proposed by Mulhern and Caprara [394], although we are not aware of widespread use of these techniques in RM. Another forecasting method proposed for RM is based on fitting historical booking to a set of cumulative booking curves. The current bookings on hand are extrapolated using these curves to give the forecast. This approach is similar in spirit to the multiplicative pick-up method discussed above.

9.3.11 Combining Forecast Methods

With computing power and storage becoming cheaper by the day, an increasingly feasible forecasting strategy is to simultaneously use several forecasting methods and pick the "best" one. Of course, identifying which method is best becomes another forecasting exercise in itself, and there have been many proposals for such a model-picking strategy.

Moreover, it may not even be necessary to identify the best-performing method: a linear combination of the forecasts with an appropriate set of weights can turn out to be consistently superior to any one of the constituent methods. This idea was proposed in an article by Bates and Granger [30] and subsequently much investigated by forecasting researchers. The intuition behind this result is that if the errors produced by two forecasting methods are negatively correlated, then combining them will reduce the overall forecast error.

So what is the best set of weights for such a linear combination? This can be determined by finding weights that minimize the mean-squared error of the combined forecast. Although it is difficult to obtain such weights analytically, various heuristics have been proposed (see the Notes and Sources of this chapter for references). The weights themselves can adapt to fresh data and be updated from period to period.

We give one set of weights proposed by Bates and Granger [30] to combine forecasts from two different models. Let MSE_i be the mean-squared error of model $i, i = 1, 2$. Let ρ be the coefficient of correlation between the errors in the forecasts of the two models. Then define the weights as α and $(1 - \alpha)$, where α is given by:

$$\alpha = \frac{MSE_2 - \rho\sqrt{MSE_1}\sqrt{MSE_2}}{MSE_1 - MSE_2 - 2 - \rho\sqrt{MSE_1}\sqrt{MSE_2}}.$$

Then the combined forecast is given by

$$\hat{Z} = \alpha\hat{Z}_1 + (1 - \alpha)\hat{Z}_2.$$

Another combination scheme, this time using adaptive weights that vary over time, is to set $\alpha = \alpha(t)$, at time t, where

$$\alpha(t) = \sum_{i=1}^{t} \frac{MSE_2(t)}{MSE_1(t) + MSE_2(t)},$$

where $MSE_i(t)$ is the mean squared error of model i at time t. The interested reader should consult Montgomery et al. [388], Gupta and Winston [230], and Foster and Vohra [192] for other similar rules and their properties.

9.4 Data Incompleteness and Unconstraining

We next look at forecasting from data that is either missing or partially observable, a common situation in RM. Indeed, once a product is closed or capacity is sold out, we normally stop observing demand at that point because most reservation systems record only actual bookings and not "attempted bookings." Ignoring this censoring can cause a significant bias in the forecasts, For instance, consider a product that had been closed consistently in the past. Its observed demand would be uniformly zero, and a forecast based on this data would forecast demand as zero. However, if the optimization system had opened this product, a positive demand might have been observed.

Incompleteness can occur in price-based RM when sales (and no-sales) are not directly observable. This can make it difficult to obtain complete information on customer purchase behavior. For example, if a customer decides not to purchase because some alternative is not available in the retail store, this information frequently goes unrecorded. Ignoring these lost sales can lead to a bias in the forecasts if the data is not corrected to account for the missing information.

Of course, companies that sell directly through their own call centers or websites have the potential, in theory, to capture attempted reservations or no-purchase outcomes. However, in our experience few actually do. And given the significant role that third-party reservations systems and distribution channels play in many RM industries, the problem of incomplete data remains an important one in RM forecasting.

Fortunately, there are several good methods available for correcting for incomplete data, which we discuss here. Our description of these methods is focused primarily on quantity-based RM because this is where the incomplete-data problem is most acute. However, the techniques are also used for estimating parameters in price-based RM, such as when correcting for stock-outs or unobservable heterogeneity in retail RM.

9.4.1 Expectation-Maximization (EM) Method

The *expectation-maximization (EM) method* is the most widely used method for correcting for constrained data in quantity-based RM. While the algorithm can be described in generic form, it is easiest to understand it by looking at specific examples. Because of its importance, we give two such examples below, one for the independent fare class model and the other for the discrete choice demand model.

9.4.1.1 Unconstraining in an Independent Booking Class Model Using EM

Consider the independent-demand model of Section 2.2, in which the demand for each product is assumed to be independent of the demand for other products. Since most current quantity-based RM implementations assume independent demand for products, the method described here (or variations of it) is very prevalent in practice.

Suppose we have $M + N$ observations of bookings for a given product, $z_1, \ldots, z_{M+N}$, of which M observations are constrained because the product was closed. We ignore the time-series aspect of the observations and treat $z_1, \ldots, z_{M+N}$ as an unordered set of observations generated by an i.i.d. process. Specifically, if the time-series data has trend or seasonality, the EM algorithm cannot be applied as shown below. (Combining unconstraining with time-series forecasting is more complicated. See McGill [376].) Our goal is to find the parameters of an underlying demand distribution for these observations.

Assume that the underlying demand distribution is normal with mean μ and standard deviation σ. (The same unconstraining procedures can be applied—albeit with different formulas—for many other distribution as well.) We further assume that *all* the observations come from a common distribution and that the observations are constrained at random, i.e., they appear randomly in the sample.[9] Since we are treating the observations as unordered, assume $z_1, \ldots, z_M$ are constrained (*right censored)* at booking limits $b_1, \ldots, b_M$, so that $z_1 = b_1, \ldots, z_M = b_M$. The remaining N observations are unconstrained.

If the data were not constrained, then it would be easy to construct the complete-data likelihood function. Namely,

$$L(\mu, \sigma, M + N) = \prod_{i=1}^{M+N} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(z_i-\mu)^2}{2\sigma^2}}, \tag{9.56}$$

with the complete-data log-likelihood function given by

$$\ln L(\mu, \sigma, M + N) = -\frac{M+N}{2} \ln 2\pi - (M + N) \ln \sigma - \frac{\sum_{i=1}^{M+N} (z_i - \mu)^2}{2\sigma^2}. \tag{9.57}$$

The μ and σ that maximizes $\ln L(\cdot)$ in (9.57) are given by the closed-form solution

$$\begin{aligned} \hat{\mu} &= \frac{1}{M+N} \sum_{i=1}^{M+N} z_i \\ \hat{\sigma}^2 &= \frac{1}{M+N} \sum_{i=1}^{M+N} (z_i - \hat{\mu})^2. \end{aligned}$$

[9] In the RM context this assumption implies that there is no correlation among demand on days when the product is sold out. Strictly speaking, this assumption rarely holds in RM practice, but it is common to ignore this correlation possibility as the alternative statistical methods are considerably more complicated.

(See Example 9.5.) However, we do not know the true values of the M constrained observations $z_1, \ldots, z_M$ and therefore cannot use this procedure directly.

The EM method uses this complete-data likelihood function in an iterative algorithm with an alternating *E-step* and *M-step* (hence the name). The E-step replaces the censored data by estimates of their uncensored values using the current estimates of the mean and standard deviation. The M-step then maximizes the complete-data log-likelihood function based on this updated data to obtain new estimates of the mean and standard deviation. The procedure is then repeated until the parameter estimates converge. The advantage of this approach is that it is much easier to estimate the complete-data likelihood than it is to estimate the incomplete-data log-likelihood function. Hence, even though we have to solve the complete-data likelihood problem many times, the overall algorithm is still very efficient.

Specifically, for our normal distribution example, let $\mu^{(k)}, \sigma^{(k)}$ represent the estimates of the parameters of the normal distribution after the k^{th} iteration of the algorithm. The steps of the EM algorithm for our time series follow:

STEP 0 (Initialize): Initialize μ and σ to be $\mu^{(0)}$ and $\sigma^{(0)}$. Good candidates for these starting values are the sample mean and sample standard deviation of all the unconstrained observations.

Let $\delta > 0$ be a small number, to be used as a stopping criterion.

$$\begin{aligned} \mu^{(0)} &= \frac{\sum_{i=M+1}^{M+N} z_i}{N} \\ \sigma^{(0)} &= \sqrt{\frac{\sum_{i=M+1}^{M+N} (z_i - \mu^{(0)})^2}{N}}. \end{aligned}$$

STEP 1 (E-step): Calculate the expected value of the censored data in the log-likelihood function assuming that they come from a normal distribution X with parameters $(\mu^{(k-1)}, \sigma^{(k-1)})$. That is, for $i = 1, \ldots, M$ calculate

$$\hat{Z}_i^{(k-1)} \doteq E[X | X \geq b_i, X \sim N(\mu^{(k-1)}, \sigma^{(k-1)})]$$

and

$$(\hat{Z}_i^2)^{(k-1)} \doteq E[X^2 | X \geq b_i, X \sim N(\mu^{(k-1)}, \sigma^{(k-1)})].$$

The formulas for these conditional expectations are somewhat complex but involve simply evaluating two integrals.

Next, for each censored observation $i = 1, \ldots, M$, replace z_i by $\hat{Z}_i^{(k-1)}$ and z_i^2 by $(\hat{Z}_i^2)^{(k-1)}$ to form the complete-data log-likelihood function $Q(\mu, \sigma)$ as in (9.57). Note in this way we are simply replacing the constrained values in the log-likelihood function by their expected values given the current estimates of the mean and standard deviation.

STEP 2 (M-step): Maximize $Q(\mu, \sigma)$ with respect to μ and σ to obtain $\mu^{(k)}, \sigma^{(k)}$, yielding

$$\mu^{(k)} = \frac{1}{M+N}\left[\sum_{i=1}^{M} \hat{Z}_i^{(k-1)} + \sum_{i=M+1}^{M+N} z_i\right]$$

and

$$\sigma^{(k)} = \frac{1}{M+N}\left[\sum_{i=1}^{M}\left((\hat{Z}_i^2)^{(k-1)} - 2\hat{Z}_i\mu^{(k-1)} + (\mu^{(k-1)})^2\right)\right.$$
$$\left. + \sum_{i=M+1}^{M+N}\left(z_i - \mu^{(k-1)}\right)^2\right].$$

STEP 3 (Convergence test): IF $\|\mu^{(k)} - \mu^{(k-1)}\| < \delta$ and $\|\sigma^{(k)} - \sigma^{(k-1)}\| < \delta$, THEN STOP;
ELSE, $k \leftarrow k+1$, GOTO STEP 1.

If the expected log-likelihood is continuous in the parameters (μ and σ in our case), a result by Wu [582] shows that if the sequence of EM estimates converges, the limiting value will be a stationary point of the incomplete log-likelihood function. Whether the sequence diverges—or converges to something other than the global maximum—is more difficult to determine and depends on the characteristics of the data set. In practice, however, the EM method has proved to be very robust.

Once convergence has been achieved—say, in iteration K—the unconstrained values for $z_i, i = 1, \ldots, M$ can be taken as $E[X|X \geq b_i]$, where X is normally distributed with $\mu^{(K)}, \sigma^{(K)}$:

Example 9.16 Consider the data set of bookings in Table 9.4 from 11 Jan to 29 Jan. The data on 13 Jan, 16 Jan and 18 Jan is constrained at the booking limit 17, 22, and 15 respectively. Assume the data comes from a normal distribution. Based on the constrained data, the parameters of the normal distribution $(\mu^{(0)}, \sigma^{(0)}) = (22.526, 7.537)$.

Let C be the capacity constraint, D the demand, $z^{(k)}$ the unconstrained value at the k^{th} iteration, and $\bar{C}^{(k)} = \frac{C-\mu^{(k)}}{\sigma^{(k)}}$. Then at the $(k+1)^{\text{st}}$ iteration, replace $\bar{z}^{(k)}$ by

$$E[\bar{z}^{(k)}|\bar{z}^{(k)} \geq C; D \sim N(\mu^{(k)}, \sigma^{(k)})]$$

Table 9.4. EM algorithm iterations on constrained data.

Date	*Constrained Bookings*	*Iteration 1*	*Iteration 2*	*Iteration 3*
11-Jan	22	22	22	22
12-Jan	15	15	15	15
13-Jan	17*	23.544*	24.216*	24.275*
14-Jan	33	33	33	33
15-Jan	16	16	16	16
16-Jan	22*	25.416*	25.699*	25.724*
17-Jan	22	22	22	22
18-Jan	15*	23.579*	23.963*	23.963*
19-Jan	22	22	22	22
20-Jan	17	17	17	17
21-Jan	23	23	23	23
22-Jan	19	19	19	19
23-Jan	31	31	31	31
24-Jan	17	17	17	17
25-Jan	30	30	30	30
26-Jan	23	23	23	23
27-Jan	31	31	31	31
28-Jan	12	12	12	12
29-Jan	41	41	41	41
Mean	22.526	23.502	23.572	23.577
S.D	7.537	7.178	7.185	7.186

given by the following formula for the normal distribution

$$z^{(k+1)} = C + \frac{\sigma}{\sqrt{2\pi}} e^{-0.5(\bar{C}^{(k)})^2} - \bar{C}^{(k)} P(D_{\sim N(\mu^{(k)},\sigma^{(k)})} \geq \bar{C}^{(k)}).$$

So at the first iteration, replace 17 by 23.544, 22 by 25.416, and 15 by and 23.579. At the second iteration, replace 23.544 by 24.216, and so on. As can be seen from Table 9.4, the algorithm quickly converges (in this case; convergence is much slower in general) to $\mu = 23.577, \sigma = 7.186$.

9.4.1.2 Unconstraining in a Discrete-Choice Dynamic Model Using EM

We next consider the problem of unconstraining under the dynamic discrete-choice model of Section 2.6.2. Recall that in this model there is an arrival probability λ in each period and consumers select among the available classes according to a discrete-choice model. The RM control problem is then to decide which products to make available at each point in time. We consider here a multinomial-logit model similar to Example 9.6, where the probability that an arriving customer purchases alternative i from a set S is given by

$$P_i(S) = \frac{e^{\boldsymbol{\beta}^{\top}\mathbf{y}_i}}{\sum_{j\in S} e^{\boldsymbol{\beta}^{\top}\mathbf{y}_j} + 1},$$

where $\mathbf{y}_j$ is a vector of attributes of alternative j and $\boldsymbol{\beta}$ is a vector of parameters. The no-purchase probability is

$$P_i(S) = \frac{1}{\sum_{j\in S} e^{\boldsymbol{\beta}^{\top}\mathbf{y}_j} + 1}.$$

The difficulty here is estimating the parameters $\boldsymbol{\beta}$ and λ from purchase data. Specifically, if we have only purchase data, it is impossible to distinguish a period without an arrival, from a period in which there was an arrival but the arriving customer did not purchase. With this incompleteness in the data, the complete-data maximum-likelihood estimation procedure of Example 9.6 cannot be used.

However, we can again apply the EM algorithm to correct for the missing data. The broad strategy is the same as the one for the normal distribution case in Section 9.4.1: start with arbitrary initial estimates of the parameters $\hat{\boldsymbol{\beta}}$ and the arrival rate $\hat{\lambda}$. Then use these estimates to compute the conditional expected value of $\mathcal{L}$, $E[\mathcal{L}|\hat{\boldsymbol{\beta}}, \hat{\lambda}]$ (the expectation step). Maximize the resulting expected log-likelihood function to generate new estimates $\hat{\boldsymbol{\beta}}$ and $\hat{\lambda}$ (the maximization step), and repeat till the procedure converges.

Suppose there are T periods. Let P denote the set of periods in which customers purchase and $\bar{P}$ denote period in which there are no purchase transactions. Let $a_t = 1$ if there is an arrival in period t and $a_t = 0$ if there is no arrival. Let $j(t)$ denote the choice made by an arrival in period t. We can then write the complete log-likelihood function as

$$\begin{aligned}\mathcal{L} = &\sum_{t\in P}\left[\ln(\lambda) + \boldsymbol{\beta}^{\top}\mathbf{y}_{j(t)} - \ln(\sum_{j\in S} e^{\boldsymbol{\beta}^{\top}\mathbf{y}_j} + 1)\right] \\ &+ \sum_{t\in\bar{P}}\left[a_t\left(\ln(\lambda) - \ln(\sum_{j\in S} e^{\boldsymbol{\beta}^{\top}\mathbf{y}_j} + 1)\right) + (1 - a_t)\ln(1 - \lambda)\right].\end{aligned} \tag{9.58}$$

The unknown data are the values a_t, $t \in \bar{P}$ in the second sum. However, given estimates $\hat{\boldsymbol{\beta}}$ and $\hat{\lambda}$, we can determine their expected values (denoted $\hat{a}_t$) easily via Bayes rule:

$$\hat{a}_t \doteq E[a_t | t \in \bar{P}, \hat{\boldsymbol{\beta}}, \hat{\lambda}]$$

$$\begin{aligned}
&= P(a_t = 1 | t \in \bar{P}, \hat{\boldsymbol{\beta}}, \hat{\lambda}) \\
&= \frac{P(t \in \bar{P} | a_t = 1, \hat{\boldsymbol{\beta}}, \hat{\lambda}) P(a_t = 1 | \hat{\boldsymbol{\beta}}, \hat{\lambda})}{P(t \in \bar{P} | \hat{\boldsymbol{\beta}}, \hat{\lambda})} \\
&= \frac{\hat{\lambda} P_0(S | \hat{\boldsymbol{\beta}})}{\hat{\lambda} P_0(S | \hat{\boldsymbol{\beta}}) + (1 - \hat{\lambda})}, \qquad (9.59)
\end{aligned}$$

where

$$P_0(S|\hat{\boldsymbol{\beta}}) = \frac{1}{\sum_{j \in S} e^{\hat{\boldsymbol{\beta}}^{\mathrm{T}} \mathbf{y}_j} + 1}$$

is the no-purchase probability for an arrival in period t given $\hat{\boldsymbol{\beta}}$.

Substituting $\hat{a}_t$ into (9.58) we obtain the expected log-likelihood

$$\begin{aligned}
E[\mathcal{L}|\hat{\boldsymbol{\beta}}, \hat{\lambda}] &= \sum_{t \in P} \left[\boldsymbol{\beta}^{\mathrm{T}} \mathbf{y}_{j(t)} - \ln(\sum_{j \in S} e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{y}_j} + 1) \right] \\
&\quad - \sum_{t \in \bar{P}} \hat{a}_t \ln(\sum_{j \in S} e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{y}_j} + 1) \\
&\quad + \sum_{t \in P} \ln(\lambda) \\
&\quad + \sum_{t \in \bar{P}} (\hat{a}_t \ln(\lambda) + (1 - \hat{a}_t) \ln(1 - \lambda)). \qquad (9.60)
\end{aligned}$$

As in the case of the complete log-likelihood function, this function is separable in $\boldsymbol{\beta}$ and λ. Maximizing with respect to λ we obtain the updated estimate

$$\lambda^* = \frac{|P| + \sum_{t \in \bar{P}} \hat{a}_t}{|P| + |\bar{P}|}. \qquad (9.61)$$

This is intuitive; our estimate of lambda is the number of observed arrivals $|P|$, plus the estimated number of arrivals from unobservable periods $\sum_{t \in \bar{P}} \hat{a}_t$, divided by the total number of periods $|P| + |\bar{P}| = |T|$. We can then maximize the first two sums in (9.60) to obtain the updated estimate $\boldsymbol{\beta}^*$. Note that this expression is of the same functional form as the complete data case (9.11). The entire procedure is then repeated.

Summarizing the algorithm:

STEP 0 (Initialize): $\hat{\boldsymbol{\beta}}^{(0)}$ and $\hat{\lambda}^{(0)}$, k=0.

STEP 1 (E-step): For $t \in \bar{P}$, use the current estimates $\hat{\boldsymbol{\beta}}^{(k)}$ and $\hat{\lambda}^{(k)}$ to compute $\hat{a}_t^{(k)}$ from (9.59).

STEP 2 (M-step): Compute $\lambda^{(k+1)}$ using (9.61).
Compute $\boldsymbol{\beta}^{(k+1)}$ by solving

$$\max_{\boldsymbol{\beta}} \left\{ \sum_{t \in P} \left(\boldsymbol{\beta}^{\mathrm{T}} \mathbf{y}_{j(t)} - \ln(\sum_{j \in S} e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{y}_j} + 1) \right) - \sum_{t \in \bar{P}} \hat{a}_t^{(k)} \ln(\sum_{j \in S} e^{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{y}_j} + 1) \right\}$$

STEP 3 (Convergence test): IF $\|(\hat{\lambda}^{(k+1)}, \hat{\boldsymbol{\beta}}^{(k+1)}) - (\lambda^{(k)}, \boldsymbol{\beta}^{(k)})\| < \delta$,
THEN STOP;
ELSE $k \leftarrow k+1$ AND GOTO STEP 1.

One interesting fact is that there can be multiple pairs $(\boldsymbol{\beta}, \lambda)$ that produce the same probabilities of sales. In this case, the EM and logit estimates will find only one such pair. To take a trivial case, suppose there is only $n = 1$ fare product and that y_1 and β are scalars. The probability that we observe a sale if this fare product is open is then

$$p = \lambda \frac{e^{\beta y_1}}{e^{\beta y_1} + 1}.$$

It is clear that there are a continuum ofvalues (β, λ) that will produce the same value p. However, the maximum-likelihood estimate will identify only one such pair. This difficulty is not a fault of the EM or logit method per se; it is a reflection of the fact that—as in this simple example—there may be more than one model that produces the same purchase probabilities. In such cases, it is simply not possible to uniquely identify the model from observed data; there is, in effect, a degree of freedom that we cannot resolve.

9.4.2 Gibbs Sampling

While the EM algorithm is the most popular and widely used method for unconstraining in RM applications, there are alternative statistical methods to deal with constrained data. We briefly describe one technique here, called *Gibbs sampling,* which is part of a broader set of methods called *Markov-chain Monte Carlo (MCMC)* methods. Although not widely used in forecasting for quantity-based RM, they have found application in price-based RM (Allenby and Rossi [10], Allenby, Arora, and Ginter [8]), econometrics (Chib and Greenberg [115]), and missing-data problems (Schafer [456]).

MCMC methods simulate a (typically intractable) target distribution $f(\cdot)$ of a (multidimensional) random variable $\mathbf{Z}$ by repeatedly simulating a sequence $\{\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(k)}, \ldots\}$, where each element in the sequence depends on the previously generated element, and the limiting distribution of $\mathbf{Z}^{(k)}$ as $k \to \infty$ is the target distribution $f(\cdot)$. By generating enough of these sequences, we can reconstruct the entire distribution $f(\cdot)$.

We first describe the Gibbs sampling method in general and then apply it to the censored normal example in Section 9.4.1.1. Let a random vector Z be partitioned into J subvectors

$$\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_J).$$

Let $f(\mathbf{Z})$ be the joint distribution of $\mathbf{Z}$—-that is, the target distribution. The Gibbs algorithm is applicable whenever $f(\mathbf{Z})$ is unknown, intractable, or difficult to sample from, but all the distributions $f(\mathbf{Z}_i|\mathbf{Z}_{-i})$, for $i = 1, \ldots, J$, ($\mathbf{Z}_{-i}$ is the vector X but without the i^{th} block) have known distributions that are easy to sample from.

Let $\mathbf{Z}^{(k)} = (\mathbf{Z}_1^{(k)}, \mathbf{Z}_2^{(k)}, \ldots, \mathbf{Z}_J^{(k)})$ be the generated sample at the k^{th} iteration.

Gibbs algorithm: Repeat the following steps till convergence (the criteria for which are discussed later):

- Generate $\mathbf{Z}_1^{(k+1)}$ from $f(\mathbf{Z}_1|\mathbf{Z}_2^{(k)}, \ldots, \mathbf{Z}_J^{(k)})$
- Generate $\mathbf{Z}_2^{(k+1)}$ from $f(\mathbf{Z}_2|\mathbf{Z}_1^{(k+1)}, \mathbf{Z}_3^{(k)}, \ldots, \mathbf{Z}_J^{(k)})$
- $\vdots$
- Generate $\mathbf{Z}_J^{(k+1)}$ from $f(\mathbf{Z}_J|\mathbf{Z}_1^{(k+1)}, \mathbf{Z}_2^{(k+1)}, \ldots, \mathbf{Z}_{J-1}^{(k+1)})$

The stationary distribution of the sequence $\{\mathbf{Z}^{(k)}\}_{k=0,1,2,\ldots}$, under relatively mild conditions, can be shown to converge to the joint distribution $f(\mathbf{Z})$.

The use of Gibbs sampling for parameter estimation usually proceeds in a Bayesian framework, in which we assume a prior distribution on the parameters, and—from a practical point of view—choose a conjugate family of distributions for the parameters.

To illustrate, let's see how to apply Gibbs sampling method to estimate the unconstrained mean and variance of a sample from a censored normal distribution with unknown mean and standard deviation, μ, σ.

Assume as in the previous section that we have a sequence of $M + N$ independent observations $\{z_i\}$, where the first M observations are constrained at $b_i, i = 1, \ldots, M$. Our problem is to estimate μ and σ. It is convenient to assume that $\mu|\sigma$ (μ given σ) has a "prior" normal distribution and σ, a "prior" inverted chi-square distribution,[10] denoted by χ^{-2}. This particular choice of distributions ensures the posterior distributions of $\mu|\sigma$ and σ are normal and inverted chi-square again.

The vector $\mathbf{Z}$ is then assumed to consist of two blocks—the first, of the unknown parameters $[\mu, \sigma]$, and the second, the vector of censored observations $(Z_1, \ldots, Z_M)$. The Gibbs algorithm begins with initial values for these two subvectors. For instance, as we did in the case of the EM application, take (μ, σ) initially equal to the sample mean and standard deviation of $\{z_i\}_{i=M+1,\ldots,N}$ and set the vector $(z_1^{(0)}, \ldots, z_M^{(0)})$ equal to the vector of censored values $(b_1, \ldots, b_M)$.

At the $(k+1)^{\text{st}}$ step, generate

$$(Z_1^{(k+1)}, \ldots, Z_M^{(k+1)}) \sim N(\mu^{(k)}, \sigma^{(k)} \mid Z_1 \geq b_1, \ldots, Z_M \geq b_M)$$

as M independent draws.

Next generate new values for

$$(\mu^{(k+1)}, \sigma^{(k+1)})$$

from a normal and inverted chi-square distribution, respectively, as follows:

$$\begin{aligned} \mu^{(k+1)} &\sim N(\bar{y}, \sigma^{(k)}) \\ \sigma^{(k+1)} &\sim (M+N-1)\bar{S}^2\chi^{-2}_{M+N-1}, \end{aligned}$$

where $\bar{y}$ and $\bar{S}$ are the sample mean and standard deviation of the M generated values and the N unconstrained values:

$$[z_1^{(k+1)}, \ldots, z_M^{(k+1)}, z_{M+1}, \ldots, z_{M+N}].$$

This procedure is repeated until the distributions of μ, σ, $z_1, \ldots, z_M$ reach stationarity. However, testing for stationarity of a distribution can be problematic (Section 9.3.2.1), so in practice a number of heuristic termination criteria are used [456]. The resulting expected value of μ and σ can then be used as our parameter estimates.

9.4.3 Kaplan-Meir Product-Limit Estimator

The Kaplan-Meir product-limit (PL) estimator ([289]) is another approach to censored-data estimation. Its origins lie in survival analysis (with continuous distributions), but here we present it in terms of censored demand observations. It is a nonparametric method, the output of

[10] A random variable Y has an inverted chi-square distribution if Y^{-1} has a chi-square distribution.

which is an estimate of the complete distribution (as in Gibbs sampling) rather than the parameters of an assumed distribution.

As before, assume we have M+N observations $z_1, \ldots, z_{M+N}$, with the first M being constrained (right-censored) at the values $b_1, \ldots, b_M$. So the observations are of the form $z_i = \min(Z_i, b_i)$, where b_i is the booking-limit (called the *limits of observation;* the event $Z_i > b_i$ is called a *loss*). As earlier, Z_i is considered independent of b_i. The *survival function* of Z_i is defined as $G(z) = P(Z_i > z)$, and an estimate of it is equivalent to an estimate of the distribution of Z.

The PL estimate $\hat{G}(z)$ of the survival function is then given as follows. List and label the $M + N$ observations in order of increasing magnitude, so that $0 \leq z_{(1)} \leq z_{(2)} \leq \cdots \leq z_{(M+N)}$. For a particular value z, let $S_z = \{r | z_{(r)} \leq z, z_{(r)} < b_{(r)}\}$. That is, S_z is the set of indices in the ordered list that are not constrained by the booking limits and have values less than z. Then

$$\hat{G}(z) = \prod_{r \in S_z} \frac{N - r}{N - r + 1}, \tag{9.62}$$

where each term above is an estimate of the conditional probability that the demand exceeds x_r given that it exceeds x_{r-1}. The main idea behind Kaplan-Meir estimate is best explained via a simple example:

Example 9.17 Suppose that we have four observations with bookings {5, 10*, 11, 18}, where the superscript * signifies a constrained observation. Suppose we are interested in the probability that $Z \geq 15$. If we ignore the constrained observation (that is, base our estimate on the unconstrained reduced sample), we get an estimate of 1/3 (one of the three unconstrained values exceeds 15).

However, we can also view $P(Z \geq 15)$ as equal to $P(Z \geq 15 | Z \geq 10) P(Z \geq 10)$. Then we estimate $P(Z \geq 10) = 3/4$ (based on the full sample) and $P(Z \geq 15 | Z \geq 10) = 1/2)$ (based on the sample of last two observations), and we obtain $P(Z \geq 15) = 3/8$. So the estimate of $P(Z \geq 10)$ helps in obtaining a better estimate of $P(Z \geq 15)$.

Kaplan and Meier show that the estimator $\hat{P}(z)$ in (9.62) gives the distribution that maximizes the likelihood of the observations. The curve given by (9.62) is remarkably easy to compute and makes no parametric assumptions. However, it can be inefficient (Miller [384]) and difficult to compare by eye (Efron [173]), and it is also difficult to compute confidence intervals for a Kaplan-Meier estimator.

9.4.4 Plotting Procedures

A hybrid parametric/nonparametric approach to censored data is based on simply fitting a parametric distribution to an nonparametric survivor function estimate $\hat{G}(z)$, such as derived using the Kaplan-Meir estimator. Such methods are called *plotting procedures*, because they correspond to plotting an empirical distribution and then inferring parameters from this plotted distribution.

To take a simple case, if the distribution is assumed to be exponential, so that $P(Z > z) = e^{-\lambda z}$, then we have that

$$\ln(G(z)) = -\lambda z.$$

Hence, if we plot the empirical function $\ln(\hat{G}(z))$, it should roughly be linear, with slope equal to $-\lambda$. One could estimate this slope via linear regression, for example. In the case of a normal distribution with mean μ and standard deviation σ, the distribution is

$$F(z) = \Phi(\frac{z-\mu}{\sigma}),$$

where $\Phi(z)$ is the standard normal distribution. Hence,

$$\Phi^{-1}(1 - S(z)) = \frac{z-\mu}{\sigma},$$

where $\Phi^{-1}(\cdot)$ is the inverse of the standard normal distribution. Therefore, by plotting $\Phi^{-1}(1-\hat{S}(z))$ we should expect to see roughly a straight line with slope $1/\sigma$ and intercept $-\mu/\sigma$. Again, values for the slope and intercept can be determined using linear regression.

While somewhat less rigorous in a strict statistical sense than other censored-data methods, plotting procedures can be attractive in practice because they are simple and intuitive.

9.4.5 Projection-Detruncation Method

The *projection-detruncation method* is similar in spirit to the EM algorithm. It has been used in the PODS simulations for quantity-based RM and its origin is credited to Hopperstad ([256, 42, 587]).

The variation over the EM method of Section 9.4.1.1 is that in the k^{th} E-step of the algorithm, instead of replacing the constrained values by an estimate of the conditional mean

$$\hat{Z}_i^{(k-1)} = E[X|(\mu^{(k-1)}, \sigma^{(k-1)}), X \geq b_i],$$

it replaces the values by the solution $\hat{Z}_i^{(k-1)}$ of the following equation

$$\int_{\hat{Z}_i^{(k-1)}}^{\infty} f(x|(\mu^{(k-1)}, \sigma^{(k-1)}))dx = \tau \int_{b_i}^{\infty} f(x|(\mu^{(k-1)}, \sigma^{(k-1)}))dx, \tag{9.63}$$

where τ is a fixed constant throughout the algorithm. While there is no formal theoretical justification of (9.63) or a proof of convergence, the heuristic interpretation is as follows. Note that (9.63) can be written

$$P\left(Z_i > \hat{Z}_i^{(k-1)} \middle| \, Z_i > b_i, \ \mu^{(k-1)}, \ \sigma^{(k-1)}\right) = \tau.$$

So $\hat{Z}_i^{(k-1)}$ corresponds to selecting a fixed fractile of the conditional distribution given the current parameter estimates $\mu^{(k-1)}, \sigma^{(k-1)}$. For example, selecting $\tau = 1/2$ would correspond to estimating $\hat{Z}_i^{(k-1)}$ as the *median* of the conditional distribution, whereas the EM method uses the *mean* of the conditional distribution. Hence, by using a small τ value the constrained observations are unconstrained more aggressively than may be the case in the EM method. Whether this leads to more accurate estimation of the mean or a faster convergence than the EM algorithm is not known, however. Zeni [587] gives an example comparing the estimates of the two methods for $\tau = 0.15$, and the estimate of the mean of the projection-detruncation method is nearly 10 percent higher than that given by the EM algorithm, though one can arguably attribute this to the choice of τ.

9.5 Error Tracking and System Control

As mentioned, all forecasts are subject to some degree of error. Hence, understanding and responding correctly to forecast errors are important tasks in practice. Here we review the main methods for error tracking and system control.

A forecaster needs to consider several types of errors. The difference between the observed data and a model fit to this data is called the *estimation error.* Such error could be due to many factors: natural randomness in the demand process, unobservable characteristics of the products or demand, mispecifications, unrealistic model assumptions such as independence of the variables or error terms. We group all such errors—errors in the estimation of the parameters of the model or the specification of the model—as estimation errors.

Forecasting error, on the other hand, is the difference between a model's predictions for a *future* observation and the subsequent observation. The difference between forecast and estimation errors is a matter of timing. Large estimation errors might compel us to refine the model or "fix" it in some way *now* because we are aware of the errors. Forecasting errors, on the other hand, are unknown at the time of the model specification and are realized only over time. There is also a dynamic, online aspect to forecasting error and system control that is distinct from the one-shot nature of estimation.

It is natural to suppose that a model that fits historical data well that, say has low estimation errors, will also generalize well and give low forecast errors. This, however, is not the case. As we show in Section 9.5.1.4, it is not uncommon to fit a model to give near-zero estimation errors based on observed data, but then find that it has atrocious predictive power. Indeed, forecasting can be said to be the art of understanding estimation errors (their sources and reasons) and then selecting and training a model properly for optimum prediction power.

9.5.1 Estimation Errors

We first look at issues involved in analyzing estimation errors—in particular, bias, specification error, model-selection criteria, and overfitting.

9.5.1.1 Bias Detection and Correction

Bias in the parameter estimates of a model is called *estimation bias.* This could arise because of the lack of a good estimator, incomplete data, or nonconvergence of the estimation procedures. A bias in the parameter estimates of a model leads to a bias in the forecasts, and in general, it is desirable to eliminate it. If the cause of the bias were known, we would, of course, fix the bias by eliminating the cause, but this is not always possible—for lack of development time, investigation time, or data, and so on. If this is the case, a simple and general method for correcting for parameter bias is the so-called *jackknife estimator* (Quenouille [431]; Tukey [519]), which we describe next.

Suppose θ is a parameter and $\hat{\theta}$ an estimator of the parameter based on an i.i.d. sample $Z_1, \ldots, Z_N$. Suppose that $\hat{\theta}$ is a biased estimator of the following form

$$E[\hat{\theta}] \quad = \quad \theta + a_1/N + O(1/N^2).$$

That is, a order $1/N$ term and a second-order error term. The jackknife estimator is calculated as follows. Let $\hat{\theta}_{-i}$ be the estimator $\hat{\theta}$ applied to the sample with the i^{th} observation removed. Define

$$\tilde{\theta}_i = N\hat{\theta} - (N-1)\hat{\theta}_{-i}, \quad i = 1, \ldots, N.$$

Define the (first-order) jackknife estimator as

$$\tilde{\theta} \quad = \quad \frac{\sum_{i=1}^{N} \tilde{\theta}_i}{N} = N\hat{\theta} - (N-1)\frac{\sum_{i=1}^{N} \hat{\theta}_i}{N},$$

which has the rather nice property that

$$E[\tilde{\theta}] \quad = \quad \theta + O(1/N^2).$$

Higher-order jackknife estimators can be defined that eliminate higher-order biases. Besides bias correction, the jackknife is a valuable tool for interval estimation and has connections to bootstrap methods (Miller [385]; Davison and Hinkley [148]).

While bias is usually undesirable, biased estimators may occasionally be beneficial if they lead to lower variance (more efficient) estimates. To give an example, if some of the explanatory variables in a linear regression are correlated (multicollinearity), the coefficients of the regression will have a high variance. A method for reducing this variance is *ridge regression,* which minimizes an objective consisting of the sum of the variance of the parameter estimates and the bias squared, so a small amount of bias is deliberately accepted (Judge et al. [273]).

9.5.1.2 Specification Errors

Specification errors are errors resulting from flawed model assumptions; that is, errors arising from a model that does not reflect the underlying data-generating process. In short, how can we be certain that the function ζ used (9.2) is indeed the "right" function to use, both explaining observed values of Z as well as providing good predictive power for future observations? Managerial judgment, visual inspection, data analysis, and statistical tests all play a role in answering this question.

Specification tests are designed to test whether a given model and its corresponding assumptions are correct. Failure to pass such a test could mean one of the following: the functional form is inadequate to represent the data-generating process; the functional form is correct, but the wrong set of independent variables have been used in the model; both the functional form and variable choice are correct, but the error term distribution is misspecified; or assumptions on the error term of the model (such as homoscedasticity or independence of errors) are violated.

There are several tests to check for misspecification (see also Section 9.2.2). The simplest ones are graphical, such as plotting values of the empirical distribution against the fitted distribution to look for a straight-line relationship, or Q-Q plots, in which the quantiles of the theoretical distribution are plotted on the x-axis and the ordered fractions of the observed values on the y-axis (a good fit is when all the values are along the diagonal). Testing an empirical distribution against a given theoretical distribution can be done using statistical procedures such as the Kolmogorov-Smirnov test. We refer the reader to DeGroot [151], pp. 554–559 for details on such tests.

Coefficient of Determination for Regressions The statistic most widely used in regressions to measure goodness of fit is the *coefficient of* *determination* (R^2), defined as follows for N observations:

$$R^2 = 1 - \frac{\sum_{j=1}^{N}(z_j - \hat{z}_j)^2}{\sum_{j=1}^{N}(z_j - \bar{z})^2}, \tag{9.64}$$

where z_j are the observations, $\hat{z}_j$ is the estimate for observation j based on the estimated parameters, and $\bar{z}$, is the mean of N observations. The R^2 value varies between 0 and 1 and signifies the percentage of the total variation in the dependent observations explained by the regression relationship. Thus, a high value of R^2 is desirable. Most commercial statistical programs (SAS, SPSS, R, S, IMSL, MINITAB, Statistica, and so on) compute this statistic automatically.

However, the choice of functional form is important, and one should not rely on quantitative measures alone. A forecaster's business intuition about the relationships and causal variables ought to play as big a role as formal statistical tests. A good R^2 value or a good visual fit does not imply a regression has good explanatory power, as we discuss below in Section 9.5.1.4 on overfitting.

The statistics of regression is concerned with many more issues than just estimating parameters and calculating R^2 values. Statistical tests exist for determining which of the independent variables is redundant, their degree of importance in determining the independent variable, their goodness of fit to the functional form, the appropriateness of the functional form and the assumptions on the errors, and so forth. For example, if the parameter estimates are assumed to be normal, then a t-test can be used to determine if the estimate is within a given interval about the true parameter value with a certain level of confidence. Similarly, a F-test can be used to test if some of the parameters are effectively redundant (values close to zero) and can be eliminated. The details of such tests are beyond the scope of this chapter, but these tests are standard and described in most statistics or econometrics texts (Kvanli et al. [318]; Judge et al. [273]; Draper and Smith [161]; Guttman [231]; Neter and Wasserman [403]).

Tests Against an Alternate Specification One form of a specification test is to test a null hypothesis that a given specification is correct against an alternate (usually more general) specification hypothesis. Depending on the type of null hypothesis, there are three classical specification tests one can use: *likelihood ratio* (LR), *Wald,* and the *Lagrange multiplier* (LM) tests. We describe only the LR test here.

Let $\boldsymbol{\beta}$ denote the vector of model parameters. Let the null hypothesis H_0 be that $\boldsymbol{\beta} \in \Omega_0$ and the alternate hypothesis be that $\boldsymbol{\beta} \in \Omega$, where typically $\Omega_0 \subset \Omega$. Then the likelihood of the observed data is as defined in (9.8). The likelihood ratio is

$$\theta = \frac{\sup_{\boldsymbol{\beta} \in \Omega_0} \mathcal{L}(\boldsymbol{\beta})}{\sup_{\boldsymbol{\beta} \in \Omega} \mathcal{L}(\boldsymbol{\beta})}. \tag{9.65}$$

If this ratio is small, the null hypothesis is rejected. That is, there is a significant loss of likelihood by restricting the parameter set to Ω_0. One attractive feature of the likelihood ratio is that the statistic

$$LR = -2 \ln \theta \tag{9.66}$$

is asymptotically χ^2 distributed, and this fact can be used for hypothesis testing.

Tests for Misspecification In contrast to the tests in the previous section, a test for misspecification does not specify a single alternate hypothesis. Instead, the null hypothesis is that the specification is correct and the alternate hypothesis is that there is a misspecification. Naturally, this is appealing as we are testing against a large number of alternative specifications using a single test. We describe next, informally, a general misspecification test strategy due to Hausman (Hausman [245]; also attributed to Durbin [169] and Wu [583]). We illustrate it by applying it to testing the IIA property in a discrete-choice model (Section7.2.2.3).

To describe the idea behind the Hausman test, consider a specification test as in the previous section. The null hypothesis H_0 is that a given specification is true; the alternate hypothesis H_1 is that another specification is true. Let $\hat{\boldsymbol{\beta}}_0$ be a consistent and asymptotically efficient estimator achieving the Cramer-Rao bound on the variance of the parameters (Section 9.2.1.3) of the specification under H_0. (In most cases, there would exist such an estimator if the null hypothesis were true; for instance, the maximum-likelihood estimators are consistent and asymptotically efficient [220] under some mild regularity conditions.) If instead H_1 were true, then $\hat{\boldsymbol{\beta}}_0$ will be biased and inconsistent under H_1 (provided H_0 and H_1 are sufficiently different and assuming that the specification of H_1 uses the same vector of parameters as that of H_0). Let $\hat{\boldsymbol{\beta}}_1$ be some other estimator for the specification of H_0—consistent but asymptotically inefficient under H_0, but consistent under H_1 also. If such estimators exist, then one can construct a test statistic out of the difference $\hat{q} = \hat{\boldsymbol{\beta}}_0 - \hat{\boldsymbol{\beta}}_1$, as this difference should be approximately centered at zero.

Hence, to test for misspecification when there is no alternate specification, one can proceed by choosing two distinct estimators for the null hypothesis specification—one efficient and one not efficient but more robust (consistent even under a mispecification) than the first one. Then, if the model is correctly specified, the difference between the estimators will very likely have a mean away from zero. To apply the statistic, the variance of $\hat{q}$, $V(\hat{q})$ has to be calculated, which fortunately turns out to be equal to the difference of the variances of $\hat{\beta}_0$ and $\hat{\beta}_1$. The test statistic used is $\hat{q}^{\mathsf{T}}V(\hat{q})^{-1}\hat{q}$, which can be shown to have an asymptotically χ^2 distribution (Hausman [245]; MacKinnon [352]). With no misspecification, $\hat{q}$ will tend to 0 w.p.1.

This specification test strategy, called the *Hausman-type test,* is quite general and has found many applications in econometrics. We illustrate the test by an example relevant to RM and price-response estimation.

Example 9.18 (HAUSMAN-MCFADDEN SPECIFICATION TEST FOR THE MNL DISCRETE-CHOICE MODEL ([244])) Given a set of observations of choices among n alternatives made by a population of N individuals, we would like to know if the MNL model is the correct specification for the choice process. Assume that the no-purchase choices are also observed.

Recall that the MNL model is characterized by the IIA property (Section 7.2.2.3): the ratio of the probabilities of choosing any two alternatives is independent of the attributes or the availability of a third alternative. Let $\mathcal{N} = \{1, \ldots, n\}$ be the set of alternatives, the probability of choosing alternative i is given by (7.6)

$$P_i(\mathcal{N}) = \frac{e^{\boldsymbol{\beta}^{\mathsf{T}} \mathbf{y}_k}}{\sum_{j \in \mathcal{N}} e^{\boldsymbol{\beta}^{\mathsf{T}} \mathbf{y}_j}},$$

where $\mathbf{y}_j$ is the M-vector of attributes and relevant characteristics of the decision maker for alternative j, $j = 1, \ldots, n$, and $\boldsymbol{\beta}$ is a M-vector of parameters to be estimated (assumed to be jointly normal with a covariance matrix $\boldsymbol{\Sigma}$).

If S a subset of the alternatives, $S \subset \mathcal{N}$, then if the IIA property holds, for $i \in S$,

$$P_i(\mathcal{N}) = P_i(S)P_S(\mathcal{N}), \tag{9.67}$$

where

$$P_S(\mathcal{N}) = \sum_{j \in S} P_j(\mathcal{N}).$$

If the IIA property fails to hold, there has to be a set 5 where (9.67) fails to hold. So if we restrict our population to customers who purchased only in S, we obtain an estimate $\hat{\boldsymbol{\beta}}_S$ based only on this data, with its covariance matrix estimated by $\hat{\boldsymbol{\Sigma}}_S$. Let $\hat{\boldsymbol{\beta}}_{\mathcal{N}}$, $\hat{\boldsymbol{\Sigma}}_{\mathcal{N}}$ be the corresponding estimates for the full choice set.

Note that there may be some elements of the M-vector of parameters that may not be identifiable from data restricted to purchases in S (for instance, alternative-specific variables where the alternatives are not in S). If such is the case, we have to restrict ourselves to a subvector corresponding to explanatory variables that vary within S, but for simplicity, assume that this subvector coincides with the full M-vector of explanatory variables.

The Hausmann specification test is based on the difference $\hat{q} = \hat{\boldsymbol{\beta}}_{\mathcal{N}} - \hat{\boldsymbol{\beta}}_S$. If the IIA property holds, the two estimates $\hat{\boldsymbol{\beta}}_{\mathcal{N}}$ and $\hat{\boldsymbol{\beta}}_S$ should coincide, and $\hat{q}$ will be a consistent estimator of 0. Then if $\mathbf{V}(\hat{q})$ is the variance-covariance matrix of $\hat{q}$ ($\mathbf{V}(\hat{q}) = \hat{\mathbf{\Sigma}}_{\mathcal{N}} - \hat{\mathbf{\Sigma}}_{\mathcal{S}}$), the test statistic

$$\hat{q}^{\mathrm{T}}\mathbf{V}(\hat{q})^{-1}\hat{q}$$

is asymptotically χ^2 distributed with degrees of freedom given by the rank of $\mathbf{V}(\hat{q})$. The null hypothesis can then be accepted or rejected with a specified degree of confidence. In principle, this has to be tested for all possible subsets S of $\mathcal{N}$. Also, there is no guarantee that the variance-covariance matrix $\mathbf{V}(\hat{q})$ is invertible. Hausmann and McFadden report that the test is not very powerful unless deviations from MNL are substantial.

9.5.1.3 Model Selection

Model selection is one of the most subtle tasks in estimation. There are no clear-cut rules; intuition, judgment, experience, and repeated testing are required to find a model that generalizes well and has good predictive power. We have already seen one iterative process for choosing a model—the Box-Jenkins methodology of Section 9.3.5 for time-series models. In this section we present additional statistical guidelines, less elaborate than Box-Jenkins, for selecting a model.

Formally, these are decision rules for selecting one of K possible models $M_1, \ldots, M_K$. The models can be time-series models or regression models or others, each with a set of parameters that we assume are estimated by a maximum-likelihood procedure. Let $\mathcal{L}^*(\boldsymbol{\beta}_k)$ represent the maximum-likelihood of model M_k based on the N observations $z = \{z_1, \ldots, z_N\}$, where $\boldsymbol{\beta}_k$ is the parameter vector of model M_k of dimension m_k.

Selection Criteria The simplest way to select a model is to rank the models according to some goodness-of-fit criterion and choose the highest-ranking one. Various decision rules have been proposed to do this, the two classical ones being the following:

- **Bayes information criterion (BIC)** The BIC of model k is defined as

  $$BIC_k = -\ln \mathcal{L}^*(\boldsymbol{\beta}_k) + \frac{1}{2} m_k \ln N,$$

  and the best model is the one with the smallest BIC.

- **Akaike information criterion (AIC)** The AIC of model k is defined as

  $$AIC_k = -\ln \mathcal{L}^*(\boldsymbol{\beta}_k) + m_k,$$

  and the best model is the one with the smallest AIC.

A number of competing criteria have been proposed, including the Fisher information criterion (FIC) [442, 559], cross-validation (CV) [492, 12], final prediction error (FPE) [464], generalized information criterion (GIC) [435].

Bayesian Selection Both the BIC and AIC have theoretical roots in the Bayesian model-selection methodology, which we describe next. Let $f_k(z|\beta_k)$ be the density function of model M_k. Let $g(\beta_k)$ be the prior distribution of the parameters of model k.

Given the data, which model is most likely? By Bayes formula,

$$P(M_k|z) = P(z|M_k)\frac{P(M_k)}{P(z)},$$

where $P(z|M_k)$ is the likelihood function for model M_k with the prior $g(\beta_k)$. Consider the posterior odds of a model M_i over M_k:

$$\underbrace{\frac{P(M_i|z)}{P(M_k|z)}}_{\text{Posterior odds}} = \underbrace{\frac{P(z|M_i)}{P(z|M_k)}}_{\text{Bayes Factor } B_{ik}} \times \underbrace{\frac{P(M_i)}{P(M_k)}}_{\text{Prior odds}}.$$

The Bayes factor indicates whether model M_i is preferred to model M_k; if B_{ik} is > 1, then M_i is preferred.

Varying i and summing over $i = 1, \ldots, K$, we get the posterior probability of model M_k as

$$P(M_k|z) = \left(\sum_{i=1}^{K} \frac{P(M_i)}{P(M_k)} B_{ik}\right)^{-1}.$$

Computing the Bayes factors can be difficult in practice, as calculating $P(z|M_k)$ involves multiple integration over the prior density $g(\beta_k)$:

$$P(z|M_k) = \int_{\beta_k} f(z_1, \ldots, z_n|\beta_k) g(\beta_k) d\beta_k.$$

One alternative is to use a holdout sample to get estimates of β_k and then use $P(z|M_k(\hat{\beta}_k))$ instead of computing the integral explicitly. The prior distribution $g(\cdot)$ is typically also calculated from a holdout sample.

Variable Selection Another task in model selection is deciding, within a given model class, which variables should be included. It is generally undesirable to include too many variables. Correlations among independent variables can lead to erroneous coefficient estimates, as in the phenomenon of multicollinearity in linear regression. Even if the explanatory variables are independent, the principle of *Occam's razor*[11]

[11] The Occam's razor principle of scientific investigation states that if E represents the evidence and $P(H|E)$ the probability of a specified hypothesis H given the evidence, if

$$P(H_1|E) = P(H_2|E) = \cdots = P(H_k|E),$$

prescribes that one should make do with as few variables as possible to achieve a given level of predictive power.

Formally, given a choice of M possible explanatory variables $Y_1, \ldots, Y_M$, which is the best subset to use? We can use the model-selection criteria discussed above (AIC, BIC, FIC), treating each subset choice as a different model. However, for large M this is computationally quite burdensome as there are $O(2^M)$ possible combinations of variables. A simpler methodology, often employed in practice, is to begin with an initial subset and then try adding one variable at a time—testing to see if it increases some measure of predictive power. Similarly, one can begin with a full set and remove one variable at a time, testing for loss of predictive power at each step. See Miller [382] for a comprehensive treatment of subset selection procedures.

More sophisticated search techniques for variable subset selection, based on hierarchical Bayes models and Gibbs sampling, have also been proposed (Mitchell and Beauchamp [387]; George and McCulloch [207]).

9.5.1.4 Overfitting

In this section we look at a common problem with fitting a model to training data—namely, *overfitting*. Rather than discuss it generally, we illustrate the problem of overfitting with an example.

Consider a set of data that is generated by the following formula (unknown to the forecaster):

$$Z_t = 0.2\cos(2\pi t/10) + 0.5\sin(2\pi t/10) + \xi_t. \tag{9.68}$$

If we perform a nonlinear regression on the first 10 points using a 10^{th} degree polynomial of the form $\zeta(t) = \sum_{i=0}^{10} w_i(t/10)^i$, we obtain the fit shown in Figure 9.12. This on surface appears to fit the data well. A cubic polynomial fit to the same data set does not providing as exact a fit on the first 11 points. However, using the formula for the 10^{th} degree polynomial for forecasting is disastrous; for instance, its projection for the 12^{th} data point is -21.77, while the actual value is 0.66, and the accuracy of projections further in the future is even worse. The cubic polynomial, in contrast, has less forecast error. The 10^{th} degree polynomial is an over-fit; it has too many degrees of freedom (in this case, 11 parameters for 11 data points!). We are in effect "fitting our model to noise" by using it. A model is said to *generalize* well if it performs well on data that it has *not* been trained on. In forecasting, we are looking

for hypotheses $H_1, H_2, \ldots, H_k$, then the simplest of $H_1, H_2, \ldots, H_k$ is to be preferred (Kotz and Johnson [311]).

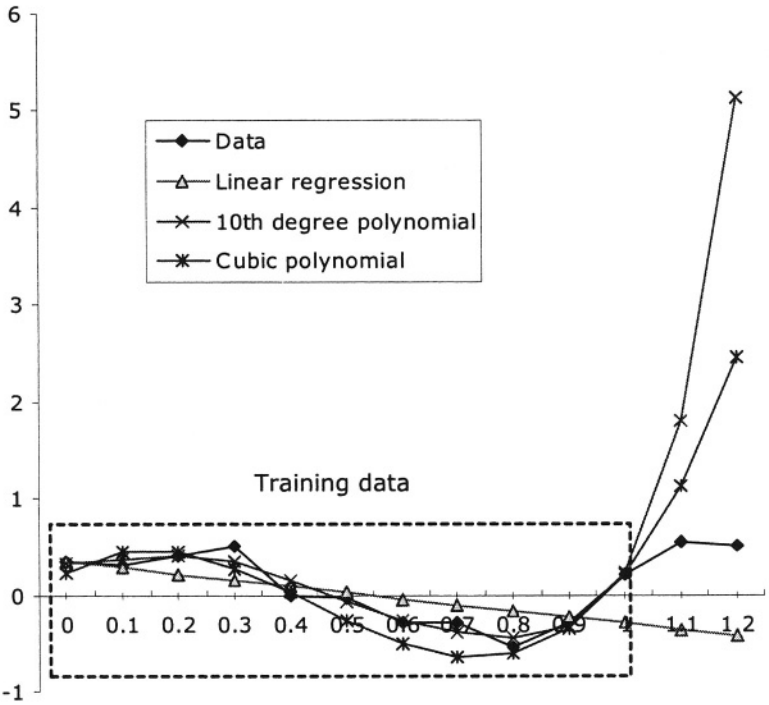


Figure 9.12. **Overfitting example. Projection of the three polynomials fit to the first 10 points of equation (9.68) (x-axis plots $t/10$).**

for good generalization properties rather than good explanations of past data.

Such overfitting problems come up during the model-selection phase for model-based methods and can be limited by considering only models that are "reasonable" from a subjective, business point of view, rather than trying blindly to find the best-fitting model based on past data. For neural networks, the problem is more subtle and difficult to detect. Because there is no explicit functional form that we choose—and because three-layer neural networks can approximate practically any function—the danger that we might overtrain and fit the network to noise is very high indeed. A good strategy to avoid overfitting is to keep a holdout sample and use the forecast errors on the holdout sample rather than on the training data to guide training.

9.5.2 Forecasting Errors and System Control

An analysis of the forecast errors is often as important as the forecast itself. Forecast error analysis is useful for the several reasons. First, the historical observed forecasting errors give a measure of the confidence one can have in the forecasting system or algorithm. Forecast errors can be used to estimate the variance in the underlying demand process and hence can be used to estimate second-order parameters of the distribution. Errors can also be used to filter out outlier data. Finally, errors can be used to track the forecast and signal unusual events or instability in the system. We look at each of these applications below.

9.5.2.1 Measures of Forecast Errors

Suppose we have been running our forecasting system for N periods and have already constructed N forecasts and made observations of the forecast relative to the actual values on these N periods. Then the forecast error for a particular period t is given by

$$e_t = z_t - \hat{Z}_t,$$

where z_t is the observed value and $\hat{Z}_t$ is the forecasted value for period t.

The following are some measures of forecast error that are used in practice:

- **Sum of forecast errors:**

$$E_N = \sum_{t=0}^{N} e_t.$$

- **Mean error:**

$$\bar{E}_N = \frac{E_N}{N}.$$

  The mean error is an estimate of the forecast bias. If the forecasting system is unbiased, the mean bias should converge to 0 as N increases.

- **Smoothed error:** This is given by the following recursive formula:

$$E_N^\alpha = \alpha e_N + (1-\alpha)E_{N-1}^\alpha,$$

  where $0 < \alpha < 1$ is a smoothing constant.

- **Mean absolute deviation (MAD):**

$$MAD_N = \frac{\sum_{t=0}^{N} |e_t|}{N}.$$

- **Mean squared error (MSE):**

$$MSE_N = \frac{\sum_{t=0}^{N} e_t^2}{N}.$$

- **Mean absolute percentage error (MAPE):**

$$MAPE_N = \sum_{t=0}^{N} \frac{|e_t/z_t|}{N}.$$

  The quantity $|e_t/z_t|$ is called the *relative error* and is not defined if z_t is 0; hence the MAPE calculation should omit such values.

- **Tracking signal (TS):**

$$TS_N = \frac{E_N}{MAD_N}.$$

It is strongly recommended that at least one of MAD, MSE, or MAPE and the TS be used to monitor a forecasting system. The primary role of MAD, MSE, and MAPE measures is to evaluate the performance of the forecasting system. Lower numbers mean better forecasts.

Among MAD, MSE and MAPE, the choice of which one to use depends strongly on the nature of the forecasts. MSE penalizes large errors for a single observation much more than MAD. Therefore, it is a better measure to detect if a few observations have large errors. If we are interested in overall performance, then MAD is generally a better choice. MAPE is useful for comparing performance across different time series, as the errors are measured relative to the data values.

9.5.2.2 Bias Detection and Correction

In addition to measuring forecast performance, a system should also monitor forecast bias. Tracking signal (TS) tests are used to monitor automated forecasts to see if the system is generating consistently biased forecasts. Typically, if the TS number exceeds a bound, an alert is generated for analysts to investigate. Most often in practice such bias is caused by a special, one-off event, but occasionally a recalibration may be required because of a fundamental change in the demand process.

There are two common tests for detecting a systematic bias in the forecast from observed errors. First, assume that the forecast is measured on a set of N observations. Let

$$\hat{T}_N = \frac{\sqrt{N}\bar{E}_N}{\sqrt{MSE_N}}.$$

Then if the forecast is unbiased, the statistic $\hat{T}$ has approximately a t-distribution with $N - g$ degrees of freedom, where g is the number of parameters in the model that are being estimated. (See Abraham and Ledolter [1], p.372.) For large N, $\hat{T}$ is approximately a standard normal (mean zero, variance one) random variable. For a given significance level, a statistical test can then be devised with the null hypothesis that the forecast is unbiased.

A second, more popular operational test for bias is to compare the absolute value of the tracking signal with a constant. (See Montgomery [388].) The forecasting system is declared biased if

$$|TS_N| > K_1.$$

The constant K_1 is usually set to be between 4 and 6. Similar tests exist using variations of the tracking signal formula, one with smoothed error in the numerator of the TS definition and a constant between 0.2 and 0.5 in the right-hand side of the bias test, and another where MSE is used instead of MAD in the denominator of the tracking signal formula and the constant in the bias test changed to be between 2 and 3.

If one knows that the forecasting system has a bias, then it would appear trivial to fix the bias—just multiply or add a correction factor. Or better still, recalibrate the system or modify the forecasting algorithms; for instance, the forecast bias could be because of a bias in the estimation of the parameters of the model (Section 9.2.1.3). But this assumes we have a precise idea of the magnitude of the bias and that it is more or less constant. As for recalibrating the model, this is often an expensive process and can involve a considerable amount of research and experimentation to come up with a better (unbiased) estimate.

9.5.2.3 Outlier Detection and Correction

Outliers are extreme values of data that are caused by corrupted records or special nonrecurring conditions in the demand process. Outlier data can severely disrupt a forecasting system. Smoothing methods—like the moving-average method—are especially susceptible to outliers because the presence of an unusual data point will distort the forecasts for several successive periods.

One technique to guard against outliers is to presmooth the data to make them more robust to the presence of outliers. The *moving-median smoothing* method in one example. Here the data is preprocessed by the following transformation:

$$\tilde{z}_t = \mathrm{Median}(z_{t-1}, z_t, z_{t+1}),$$

and the forecast is trained as if $\tilde{z}_t$ were the real data sequence. This is an example of a nonlinear smoothing method. Many such nonlinear data smoothers exist. (See Tukey [520].) Care should be exercised, however, when the data has seasonality or other periodic effects. The data may first have to be deseasonalized before using such filters.

Another technique is to try to identify and remove outliers before feeding the data to the forecasting system. One such outlier identification test is to consider a data point t an outlier if

$$\left|\frac{e_t}{MAD_t}\right| > K_2,$$

where the value of K_2 is chosen to be between 5 and 6.

9.6 Industry Models of RM Estimation and Forecasting

In this section we give some examples of specific RM forecasting models. The models are intended to be representative of those used in a particular industry to forecast a particular quantity of interest: for example, no-shows, cancellations, and groups forecasting in the airline, rental-car, and hotel industries; ratings forecasting in the media industry; sales response functions in the retail industry; promotion effects forecasting for manufacturers; and load forecasting in the electricity and gas industries. Many variations of these models are possible, and the examples presented here are intended only as illustrations—not recommendations—of forecasting approaches.

9.6.1 Airline No-Show and Cancellations Forecasting

Forecasts of cancellation and show-up rates are key inputs to the overbooking module of an airline RM system. In addition to the statistical and operational techniques discussed in this chapter so far, this example also highlights the use of data-mining algorithms for forecasting.

The first problem in cancellation forecasting is coping with reservations data. If one uses only net-bookings data for forecasts—not uncommon in RM systems—new bookings may hide cancellations. For instance, if in a period there are 100 bookings on hand, and during the period 20 new bookings are realized, but 10 current bookings cancel, then net-bookings data may make it appear that there have been 10 new bookings and 0 cancellations. Cancellation forecasts based on such data will then be biased. Similarly, go-shows or walk-ups—that is, people who show up without reservations (distinguished from regular bookings by the fact that it is lumpy demand occurring at the time of service)—

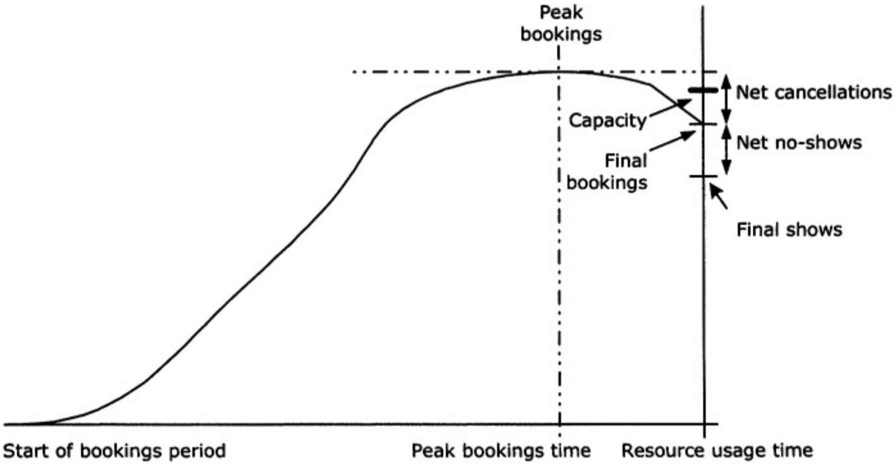


Figure 9.13. A booking curve showing net cancellations and net no-shows.

may also hide no-shows. So depending on the data being used and the requirements of the overbooking optimization, we may need to decide whether we are forecasting gross (actual) cancellations or net (observed) cancellations (Figure 9.13).

Both no-show and cancellation rates can be defined at different levels of aggregation, for the entire cabin or by fare class. Defining rates by fare class is more accurate as significant differences may exist between fare classes—for example, some may have penalties for cancelling while others may not. Cancellations can also be defined over different intervals of time, as incremental cancellations over a given period or total cancellations over the entire booking period.

Besides the level of aggregation, the cancellation rate and no-show rate can have different interpretations—(1) as the probability that a given individual booking will cancel or no-show or (2) as a fraction of the total number of bookings at a given point of time (either current time or some time in the future) that are likely to cancel or no-show. The second interpretation leads to the concept of a cancellation curve over the booking period. The cancellation rate may change over time as very early bookings tend to have higher cancellation rates than later ones (see Figure 9.14). A full cancellation curve is usually needed only in dynamic overbooking models.

For illustration, consider the binomial model of Section 4.2.1. If there are N current bookings, the cancellation rate p_C is the probability that a booking will cancel before the time of service. We define the no-show rate p_{NS} similarly. Both p_C and p_{NS} are assumed to be constant and

No-shows and cancellation forecasting can further be divided into two parts: forecasting the cancellation behavior of customers who already booked and those who will book in the future. For the former, it is possible to exploit correlations between customer cancellation probabilities and purchase characteristics like time of booking, source of booking, amount paid, cancellation penalties, and refund policies associated with the fare, to improve the forecasts.

Kalka and Weber [280], Feyen and Hüglin [190], and Westerhop [561] report airline no-show and cancellation forecasting for existing customers using data-mining and data-discovery tools on PNR data. Some of the attributes used are origin, destination, flight time, return trip, booking class, number of passengers traveling together, flight time, number of connections, and connection time. Feyen and Hüglin [190] use logistic regression on the attributes and the observed rates for prediction while Kalka and Weber [280] use induction trees. (See Quinlan [432].)

The methodology in Kalka and Weber [280] can be illustrated in Figure 9.15 for two attributes—flight time and booking class. The historical bookings and cancellations are mapped to the attribute space, and we partition the space by partitioning the ranges on the attributes. This is somewhat analogous to clustering points into groups, except that we are now interested in rules for partitioning each attribute dimension, rules that subsequently will be used to categorize new observations with its likelihood of cancellation. A cancellation probability is calculated for each box as the fraction of bookings in that box that cancel. For any new booking, its cancellation probability is derived by looking up the box it falls in and taking its corresponding cancellation probability. Data-mining tools use artificial intelligence rules-based techniques to partition the customer attribute space and construct an induction tree that gives a sequence of rules to be applied to classify observations.

9.6.2 Groups Demand and Utilization Forecasting

Bookings for units of five or more are usually classified as groups. Groups in RM can either be ad-hoc groups (one-shot groups such as school excursions or crews) or series groups (repeating groups—for example, bookings by a package-tour operator). (See Sections 10.1.2 and 10.2.1.) In this section we describe the forecasting tasks associated with groups.

Forecasting of group bookings demand is rarely done. This is because ad-hoc groups are such rare events that it makes it difficult to try to forecast demand from such sources. Series groups, on the other hand, are negotiated so far in advance that they make forecasting unnecessary.

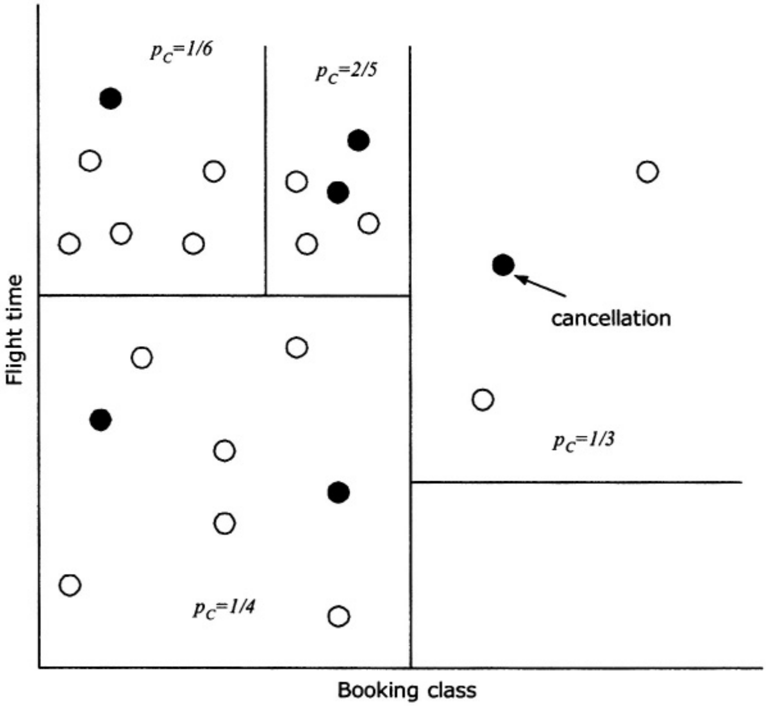


Figure 9.15. Induction tree on cancellations data.

However, group-utilization forecasting is an important task. Group utilization is defined as the percentage of a group reservation that will eventually show up. In principle, it is no different from cancellations and no-show forecasting. However, it is treated separately because groups may act as a unit, with strong correlation between the members of the unit. For instance, a group may cancel as a whole, in which case there is a sudden lumpy change in the available capacity. Because of its potential impact on availability and the higher risk involved in groups canceling as a whole, group utilization is usually tracked separately from regular cancellations and no-shows by dedicated analysts or managers.

Analysts also have better information about groups reservations than individual bookings because the reservation is usually made directly through personal contact. The source of a group reservation (such as a tour operator, cruise-line or agency) and type of group (such as a corporate meeting or convention) also helps in tracking historical usage rates. Group utilization forecasting differs if the group is an ad-hoc group or a series group. Ad-hoc groups are more likely to cancel as a whole (or not pay by the deadline), while series groups, being negotiated contracts for a long period, tend to survive till the usage date with only partial cancellations.

A forecast of group utilization is made on a historical series of utilization ratios constructed for groups with similar characteristics (same group type, group booking source, market, and so on). The forecast is usually made for each individual group reservation and updated constantly as new information and confirmations come in. Causal models are widely used to forecast group reservations because of the rich data available specific to each individual group. Group utilizations have been found to be correlated with group type (ad hoc or series), origin location of bookings, group size, penalty costs, historic cancellations, booking source, time of booking, and group travel purpose, among others. Bayesian models are also suitable because they allow analysts' beliefs on the group's utilization to be incorporated into the forecast.

9.6.3 Sell-Up and Recapture Forecasting

Sell-up and recapture are used in some RM models as discussed in Section 2.6.[12] The sell-up probability for a class is the probability that customers for that class will buy-up to at least one of the other higher classes (of the same resource) if their class is closed (this is called *differential sell-up rate* in Gorin [217]; the sell-up rate used in Belobaba and Weatherford [37] is only between the class and the next highest class). Recapture occurs when the customer buys an alternative resource (say, on a different date or time) if their requested class is closed.

There are several difficulties in estimating buy-up and recapture probabilities. For example, how do we tell if a customer is an "original" customer or a "recaptured" customer? Looking at transactional data alone, this is impossible to determine. It is common practice to pass this burden on to analysts, who are required to input buy-up and recapture probabilities for each market using their best judgment. Given the number of markets/resources/date combinations, often a single number is used for each market or for the firm as a whole.

Other approaches are based on data. Gorin [217] proposes the following formula for estimating the sell-up rate of a class for a resource: Let s_i represent total number bookings in class i (over a collection of sample historical data for the resource). Assume the classes are indexed with the lower index having a higher fare. Then Gorin [217] defines the

[12] Andersson [18] defines a customers who neither sells-up nor is recaptured but buys an alternative product from a competitor, as a *deviation*.

sell-up rate p_s as

$$p_s = \frac{(s_{i-1}|i-1 \text{ closed and } i \text{ open}) - (s_{i-1}|i-1 \text{ and } i \text{ both open})}{(s_i|i \text{ open})}.$$

However, he also states that this estimate likely is biased, so it should be used with caution. Recapture effects are not considered by Gorin.

Consumer-choice models provide a more systematic approach to buy up and recapture estimation. An early attempt at such a model is by Maynes and Wood [368], who build an econometric model of demand for three latent market segments as a function of price, schedule attributes, and competitor prices and availability. The ratio of the forecasts of demand for a class on a resource and a lower fare class on the same resource provides the sell-up probability for the lower class. This approach can be extended to estimate recapture rate as well. However, these rates are calculated on a pairwise basis only, independent of what other options are available at that time.

Andersson [19] presents a richer model of consumer behavior based on utilities and discrete-choice theory. At any given point of time, a choice set S is defined as a set of competing resource/class combinations for class j on resource i. If j is closed on i, then define S_{-ij} as the set S without class j on resource i. The estimate of the recapture rate is then defined as follows:

$p(k,l|S_{-ij})$ = Probability that resource k, class l is chosen when i, j is closed but all other choices in S_{-ij} are open.

$p(k,l|S)$ = Probability that resource k, class l is chosen when all choices in S are open.

Then the recapture rate by combination k, l from i, j, denoted $p_{i,j}(k,l)$, is defined as

$$p_{i,j}(k,l) = p(k,l|S_{-ij}) - p(k,l|S).$$

The probabilities $p(k,l|S_{-ij})$ and $p(k,l|S)$ can be estimated using an appropriate discrete-choice model. Andersson [19] (see also Köhler [308]) reports a study at Scandinavian Airlines where the choice probabilities were estimated using a MNL model fit from both transactional data and passenger surveys. The passenger surveys were in the form of games (lasting around 10 minutes), presenting alternatives of price, departure time, restrictions, and airline brand name.

9.6.4 Retail Sales Forecasting

Retail RM requires an estimate of a demand function. Besides price, advertising, product features, past sales, economic conditions, store location, brand effects, weather, and competitor actions are some factors that strongly affect sales. Consequently, in retail marketing forecasts, in contrast to airline or hotel RM, causal models are widely used.

As discussed in Chapter 7, there are two basic approaches to demand function modeling. One way of incorporating the effects of marketing variables on sales is through models of individual consumer choice behavior. Then in a bottom-up forecasting fashion, these individual choices are aggregated to get total demand. Another approach—called *aggregate forecasting*—is to model aggregate demand directly as a function of price and other marketing variables. We focus on this latter approach here, as it is prevalent both in marketing theory and practice, and as we covered discrete-choice models earlier.

Let Z denote sales, and let the marketing variables be represented by $y_1, \ldots, y_M$ for multivariate models and by y for univariate models: Consider a basic sales response model of the form

$$Z = f(y_1, \ldots, y_M) + \xi.$$

The functional forms are usually designed such that either (1) absolute change in the marketing variables leads to an absolute change in sales or (2) percentage (relative) change in the marketing variables leads to an absolute change in sales. For instance, the function $Z = \beta y$ is of the former kind (since $\partial Z = \beta \partial y$), while the function $Z = \beta \ln y$ is the latter kind ($\partial Z = \beta \partial y / y$).

The function $f(\cdot)$ can be a *static* function of variables of the current period only, or a *dynamic* function capturing the effects of marketing variables in past periods (for example, advertising done in the past month has an effect on the sales of this month). Below are some examples of static sales response functions.

- **Semilogarithmic model:**

$$Z = \beta_0 + \beta_1 \ln y_1 + \cdots + \beta_M \ln y_M + \xi. \tag{9.69}$$

Percentage sale in a marketing variable leads to an absolute change in the sales.

- **Multiplicative or power model:**

$$\ln Z = \beta_0 + y_1^{\beta_1} + \cdots + y_M^{\beta_M} + \xi. \tag{9.70}$$

The β's have the interpretation of elasticities. A more general form of (9.70) is called the *interactive* model and is given by the sum of all possible products of the variables:

$$Z = \sum e^{\beta_0} y_1^{\beta_1} \cdots y_m^{\beta_m}. \tag{9.71}$$

It is rarely used in this full form.

- **Exponential model:**

$$\ln Z = \ln Z_{\max} - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{y} + \xi. \tag{9.72}$$

Sales exhibit increasing returns to scale as the value of the marketing variable (say price) goes down to zero. $Z_{\max}$ represents the maximum possible sales.

- **Log-reciprocal or S-shaped model:**

$$\ln Z = \beta_0 - \frac{\beta_1}{y} + \xi,\ \beta_0 > 0. \tag{9.73}$$

This function possesses an inflection point at $y = \beta_1/2$. Sales show increasing marginal returns for y less than the inflection point and decreasing marginal returns from then on.

Other S-shaped curves are possible using *logistic* models such as the following *log-linear* and *double-log* models:

$$\ln(\frac{Z}{Z_{\max} - Z}) = \beta_0 + \sum_{j=1}^{M} \beta_j y_j + \xi \tag{9.74}$$

$$\ln(\frac{Z - Z_{\min}}{Z_{\max} - Z}) = \ln \beta_0 + \sum_{j=1}^{M} \beta_j \ln y_j + \xi. \tag{9.75}$$

- **Gutenberg model:**

$$Z = \beta_0 - \beta_1 \sinh[\beta_2(y - \bar{y}] + \xi. \tag{9.76}$$

$\bar{y}$ is a reference value for the marketing variable (for instance average competition price). The Gutenberg model is a complicated but flexible function. Simon [471] gives an application using this model.

Next we give some examples of dynamic sales response functions, in which the sales in a period is a function of variables of the past (lagged) periods, future (lead) customer actions or the current period:

- **Geometric distributed-lag model:** This is a dynamic model that relates the sales in period t to observed values in previous periods with exponentially decreasing weights:

$$Z_t = \beta_0 + \beta_1(1 - \alpha) \sum_{j=0}^{\infty} \alpha^j Z_{t-j} + \xi,\ 0 < \alpha < 1. \tag{9.77}$$

- **SCAN*PRO model:** This is a widely used store-sales model (proposed by Wittink et al. [572]) for determining the effect of promotions on sales. Denote, for brand i in store k in period t,

Z_{ikt}	Sales
W_t	Week-of-the-year indicators ($W_t = W_{t+52}$)
p_{ikt}	Discounted price
$\tilde{p}_{ikt}$	Nondiscounted price
F_{ikt}	0-1 indicator variable, for feature
D_{ikt}	0-1 indicator variable, for display
v_{ikt}	Inventory
ξ_{ikt}	Error term.

Then the model is given by

$$\begin{aligned} Z_{ikt} &= e^{\beta_{0ikt}} \delta_{ikt}^{W_t} \beta_{2ik}^{F_ikt} \beta_{3ik}^{D_ikt} \beta_{4ik}^{v_ikt} e^{\xi_{ikt}} \\ & \prod_{m=1,m\neq i}^{M} \left(\frac{p_{mkt}}{\tilde{p}_{mkt}}\right)^{\beta_{1mkt}} \beta_{2imk}^{F_mkt} \beta_{3imk}^{D_mkt}. \end{aligned} \tag{9.78}$$

The δ's and β's for each period have to be estimated from data.

There are literally hundreds of models such as these studied by marketing scientists, with many empirically tested on real-world data. Once a model has been fixed, regression is the most common approach for estimating static models, while time-series methods (Section 9.3.2) are common for estimating dynamic models.

9.6.5 Media Forecasting

Forecasting for broadcast media presents some unique challenges. (Media RM is discussed in detail in Chapter 10.) Prices for advertising are quoted as cost per thousand impressions. For print and television firms, the circulation and ratings determine how much the firm can charge for their advertisement space. Internet media rates are based on page-views or click-through metrics. Market-research firms such as Nielsen, IRI, and Media Metrix are dedicated to measuring the size of the circulation (print), page-views (Internet) and audience (television, radio).

A broadcaster faces two main forecasting tasks. One is to forecast ratings for shows by day-of-week and season; the other is to forecast demand for advertising slots for these shows. Forecasting the latter is usually much easier than the former because the network has knowledge of its customers—their historical preferences and buying patterns, required demographics, and in many cases, even their advertising budget. A forecast of demand is first constructed by making an estimate of each customer's demand (often manual, based on last year's demand) and then making a tentative sales plan satisfying the customer's preferences based on the ratings forecasts (Bollapragada et al. [83]). In this section, we concentrate on ratings forecasts, which is a good example of a rather difficult causal forecasting problem.

Few TV or radio managers rely on formal ratings forecasting models for their own programming decisions; surveys, gut feeling, and innate programming intuition seem to be the dominant methodologies in practice. These forecasts, though often subjective, can be helpful for RM purposes as well, as they reflect managerial judgment (for instance, they can be used to form priors in a Bayesian framework).

Recently, several methods have been proposed based on formal models of consumer viewing behavior. Television viewing habits are conceptualized as a two-stage process. In the first stage, the individual decides whether or not to watch TV. This leads to a forecast of the total aggregate TV viewing population at any given time. Once a decision to watch TV is made, the individual chooses one of the available programs, which leads to show-level ratings. (See Gensch and Shaman [206].) This two-stage model suggests using a time-series model to predict aggregate viewership by time and day of week based on recent programming data, and then a discrete-choice model to predict ratings by show. Past viewership, viewing time, seasonality, and regional differences are good predictors of aggregate viewership, while the show characteristics, slot, show-promotion, lead-ins (the popularity of the program before) and lead-outs (and the program that runs after) influence the market share of a show.

For example, Reddy, Aronson, and Stam [437] building on the work of Horen [257], use a regression model to predict the ratings of TV shows running for multiple seasons and hence with some historical data. Shows are classified into homogeneous types, based on their characteristics (movie, news, afternoon talk show). The model is:

$$\begin{aligned} Z_t^i \; = \; & \beta_0 + \sum_l \beta_l^A A_l^i + \sum_j \beta_j^S S_j^i + \sum_k \beta_k^D D_k^i + \\ & \sum_m \beta_m^{\top} T_m^i + \sum_p \beta_p^R R_p^i + \sum_{u \neq i, (u,i) \in U} \beta_i^{INT} I_u^i + \xi_{it}, \end{aligned} \tag{9.79}$$

with the following variable definitions,

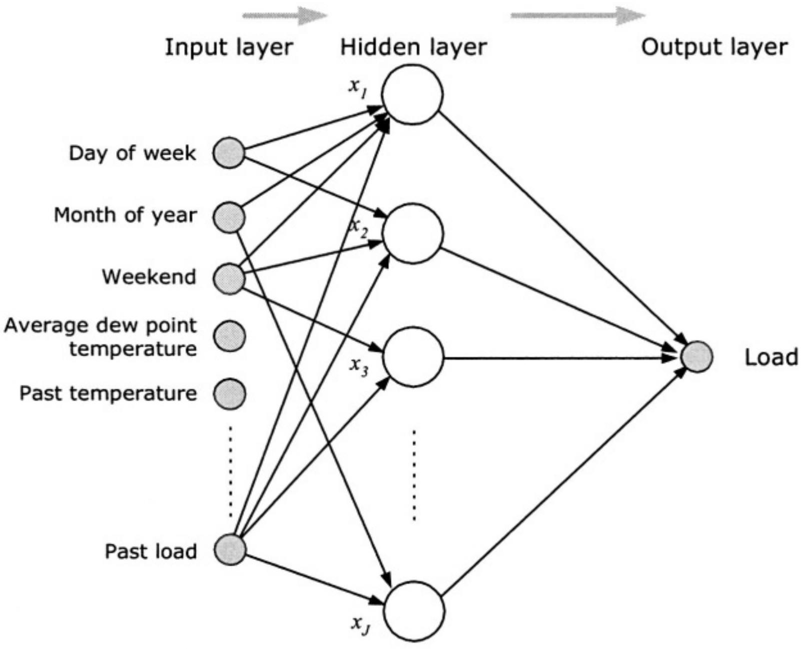


Figure 9.16. Neural network for gas-load forecasting.

Z_t^i Rating of show i in time slot t
A_l^i Measure of the relative perceived attractiveness of show i of type l (managerial rating from 0 to 10)
S_j^i 0-1 indicator variable, 1 if show i is of type j, 0 otherwise
D_k^i 0-1 indicator variable, 1 if show if show i is scheduled on day k, 0 otherwise
T_m^i 0-1 indicator variable, 1 if show i is in time slot m, 0 otherwise
R_p^i 0-1 indicator variable, 1 if show i is an hour-long show, 0 otherwise
I_u^i 0-1 indicator variable, 1 if show u (of type s) leads into show i (of type r), 0 otherwise
ξ_{it} Residual error term.

9.6.6 Gas-Load Forecasting

We next look briefly at a gas-load forecasting system using neural-network methods, reported to be implemented at Williams Gas (Lamb and Logue [327]). The model forecasts short-term (between one to five days ahead) demand for gas in a pipeline. The pipeline has thousands of meters drawing gas from it, each with variable demand. The factors that affect this demand were identified as

- Weather parameters (such as temperature, humidity, wind direction, and so on supplied by weather data vendors)

- Historic load
- Calendar (hour of the day, day of the week, holiday, month)
- Expected gas demand for each meter.
- Price (historical, current and competitor prices).

Figure 9.16 shows an example of a three-layer neural network using some of these potential inputs.

9.7 Notes and Sources

We have given many reference in the text of the chapter. Here we gather some general references on the topics of the chapter along with some additional pointers.

Regression-related topics can be found in any advanced statistic or econometric books. Here are some references: DeGroot [151] and Kvanli [318] for introductory statistics; Maddala [353], Greene [220], and Judge et al. [273] for econometrics. Some books devoted exclusively to regression are Draper and Smith [161], Guttman [231], and Neter and Wasserman [403].

Books on forecasting are available at all levels. We recommend Montgomery et al. [388] for a general introduction to operational forecasting and Harvey [242, 243] for a more advanced treatment of time-series analysis and Kalman filtering.

For books on neural networks, we recommend Bishop [69] for a very readable yet rigorous introduction to neural networks (albeit for pattern recognition) Our treatment follows also Müller, Reinhardt, and Strickland [397]. Some useful survey papers on the use of neural networks in forecasting are Poli and Jones [423], Cheng and Titterington [110], Zhang, Patuwo, and Hu [588], Hill, Marquez, O'Connor, and Remus [252], Hill, O'Connor, and Remus [253]. The application of neural networks to predict consumer choice can be found in West, Brockett, and Golden [562] and Dasgupta, Dispensa and Ghose [144].

For estimation of price-response functions and market-share models, see the following marketing science text books: Eliashberg and Lilien [174], Wedel and Kamakura [558], Hanssens, Parsons, and Schultz [235], Cooper and Nakanishi [127], Dasgupta, Dispensa, and Ghose [144], Hruska [259], West, Brockett, and Golden [562], Hill et al. [253], Zhang [588], and Lee et al. [335]. Kalyanam [283] proposes a Bayesian mixture model of pricing specifications when there is no consensus on the right model.

See Berry, Levinshohn, and Pakes [52], Berry [53], Besanko, Gupta, and Jain [63] and Chintagunta, Kadiyali, and Vilcassim [116] for estimation in a competitive market similar to the method described in Example 9.9. The problem of endogenity in estimation has received much recent attention in the marketing science literature spurred by the paper of Berry [53]. See also Chintagunta, Kadiyali and Vilcassim [116] and Villas-Boas and Winer [536] for further studies on endogeneity. Nevo [406] gives an excellent practical guide to estimating random-coefficient logit models of demand.

The Bayesian method of updating of parameters can be incorporated into many of the time-series methods of Section 9.3.2 also in a fairly straightforward manner. (See, for instance, Montgomery et al. [388].) The empirical Bayes method that we cover in this chapter is not the only possibility for handling hierarchical Bayes methods. See Lindley and Smith [345] and Blattberg and George [75] for alternatives. Hierarchical Bayes methods have also found application in modeling heterogeneity in preferences in discrete-choice models (Albert and Chib [6]; Allenby and Rossi [11]; Huber and Train [260]), and in conjoint analysis (Allenby and Ginter [9]; Lenk et al. [339]).

Literature on combining forecasts is also quite vast, given its promise of returning more than the sum of its parts. The standard references in this area are Newbold and Granger [407], Granger and Newbold [218], Makridakis and Winkler [356], Clemen and Winkler [122], Clemen [123], Gupta and Wilson [230], Schmittlein, Kimm, and Morrison [457], Morrison and Schmittlein [391], and Foster and Vohra [192]. See also Montgomery et al. ([388], p.192).

One of the few textbooks dedicated to the EM algorithm is McLachlan and Krishnan [377]. The book also contains many applications, convergence properties and lists a large number of EM references. Connections with the Gibbs method is mentioned, but the reader should refer to Schafer [456] dedicated to MCMC methods. For an introduction to Gibbs Sampling, see the article of Casella and George [101] and Gilks, Richardson and Spiegelhalter [212]. For Gibbs algorithm applied to missing-data problems, see Gelfand, Smith and Lee [203].

Both the origins of EM and Gibbs sampling (at least their ideas) can be traced far back, but Dempster et al. [152] and Geman and Geman [205] are credited with their invention and popularization.

The original paper of Kaplan and Meier [289] is still a good introduction to the Kaplan-Meier estimator. Many books on survival analysis (Miller [383]; Cox and Oakes [134]) also describe the method in detail. Logistic regression has been proposed as a parametric alternative to the Kaplan-Meier curve, with good properties and flexibility, and with all the advantages of a parametric form (Efron [173]).

The unconstraining methods described here are not the only alternatives though. It is possible to use the bootstrap, jackknife as well as regression with censored data to get estimates of the parameters of a censored sample. See Efron [171] and Davison and Hinckley [148] for an introduction to using bootstrap for unconstraining and for regression with censored data.

APPENDIX 9.A: Back-Propagation Algorithm for Neural-Network Training

We illustrate the back-propagation algorithm for training a neural network on our example of Figure 9.9.

Because we chose the linear function $\tilde{f}(h) = h$ as the activation function for input and output nodes, we represent, by a slight abuse of notation, the n^{th} instance of an input and its corresponding output of the neural net as $\mathbf{y}_n = (y_{n1}, \ldots, y_{nI})$ and $\hat{\mathbf{Z}}_n = (\hat{Z}_{n1}, \ldots, \hat{Z}_{nK})$ respectively. As always, let $\mathbf{z}^n$ be the actual observation at the n^{th} instance. Assume we are given a set of N training data instances (a set of N input-output pairs $(\mathbf{y}_n, \hat{\mathbf{Z}}_n)$, $n = 1, \ldots, N$ that we will use to determine weights of the neural network).

For training instance n, the state of node j in the hidden layer is then $x_{nj} = f(h_{nj})$, where $h_{nj} = \sum_{i=1}^{I} w_{ij} y_{ni} - \nu_j$, and the state of node k of the output layer is $\hat{Z}_{nk} = \tilde{f}(h_{nk}) = h_{nk}$, where $h_{nk} = \sum_{j=1}^{J} w_{jk} x_{nj} - \nu_k$.

For the given set of transfer functions $f(\cdot), \tilde{f}(\cdot)$, our objective is to choose the weights w_{ij}, w_{jk} and the activation threshold values ν_j, ν_k such that they minimize the squared deviation between the output values of the network and the actual observations:

$$S[w_{ij}, \nu_j, w_{jk}, \nu_k] = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} [z_{nk} - \hat{Z}_{nk}]^2.$$

The error back-propagation method performs this minimization iteratively in two stages (for a three-layer network), the first stage corresponding to the output layer and the second stage to the hidden layer. At each stage, the weights and threshold values are updated in the spirit of the steepest-descent algorithm of nonlinear optimization (see Bertsekas [59]), as follows:

STEP 0: Choose an initial set of values for the w's and ν's. Choose a step-size δ (which can either be fixed or chosen according to a step-size selection rule; see Bertsekas [59]).

STEP 1: Update the w's for arcs between the hidden layer and the output layer as follows:

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk},$$

where

$$\Delta w_{jk} = -\delta \frac{\partial S}{\partial w_{jk}} = \delta \sum_{n=1}^{N} [z_{nk} - \tilde{f}(h_{nk})] \tilde{f}'(h_{nk}) \frac{\partial h_{nk}}{\partial w_{jk}}.$$

Update the ν's of the output layer as follows:

$$\nu_k \leftarrow \nu_k + \Delta\nu_k,$$

where

$$\Delta\nu_k = -\delta\frac{\partial S}{\partial \nu_k} = \delta\sum_{n=1}^{N}[z_{nk} - \tilde{f}(h_{nk})]\tilde{f}'(h_{nk})\frac{\partial h_{nk}}{\partial \nu_k}.$$

STEP 2: Update the w's for arcs between the input layer and the hidden layer as follows:

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij},$$

where (applying differentiation using the chain rule)

$$\begin{aligned}\Delta w_{ij} &= -\delta\frac{\partial S}{\partial w_{ij}} \\ &= \delta\sum_{n=1}^{N}[\sum_{k=1}^{K}(z_{nk} - \tilde{f}(h_{nk}))\tilde{f}'(h_{nk})]w_{jk}f'(h_{nj})\frac{\partial h_{ni}}{\partial w_{ij}}.\end{aligned}$$

Update the ν's of the hidden layer as follows:

$$\nu_j \leftarrow \nu_j + \Delta\nu_j,$$

where

$$\begin{aligned}\Delta\nu_j &= -\delta\frac{\partial S}{\partial \nu_j} \\ &= \delta\sum_{n=1}^{N}[\sum_{k=1}^{K}(z_{nk} - \tilde{f}(h_{nk}))\tilde{f}'(h_{nk})]w_{jk}f'(h_{nj})\frac{\partial h_{nj}}{\partial \nu_j}.\end{aligned}$$

STEP 3: If convergence criterion is not met, GOTO STEP 1.