1
Introduction and Preview

1.1 Multivariate Analysis

This book invites the reader to learn about multivariate analysis, its modern ideas, innovative statistical techniques, and novel computational tools, as well as exciting new applications.

The need for a fresh approach to multivariate analysis derives from three recent developments. First, many of our classical methods of multivariate analysis have been found to yield poor results when faced with the types of huge, complex data sets that private companies, government agencies, and scientists are collecting today; second, the questions now being asked of such data are very different from those asked of the much-smaller data sets that statisticians were traditionally trained to analyze; and, third, the computational costs of storing and processing data have crashed over the past decade, just as we see the enormous improvements in computational power and equipment. All these rapid developments have now made the efficient analysis of more complicated data a lot more feasible than ever before.

Multivariate statistical analysis is the simultaneous statistical analysis of a collection of random variables. It is partly a straightforward extension

A.J. Izenman, *Modern Multivariate Statistical Techniques*, Springer Texts in Statistics,
DOI 10.1007/978-0-387-78189-1_1, © Springer Science+Business Media New York 2013

of the analysis of a single variable, where we would calculate, for example, measures of location and variation, check violations of a particular distributional assumption, and detect possible outliers in the data. Multivariate analysis improves upon separate univariate analyses of each variable in a study because it incorporates information into the statistical analysis about the relationships between all the variables.

Much of the early developmental work in multivariate analysis was motivated by problems from the social and behavioral sciences, especially education and psychology. Thus, factor analysis was devised to provide a statistical model for explaining psychological theories of human ability and behavior, including the development of a notion of general intelligence; principal component analysis was invented to analyze student scores on a battery of different tests; canonical variate and correlation analysis had a similar origin, but in this case the relationship of interest was between student scores on two separate batteries of tests; and multidimensional scaling originated in psychometrics, where it was used to understand people's judgments of the similarity of items in a set.

Some multivariate methods were motivated by problems in other scientific areas. Thus, linear discriminant analysis was derived to solve a taxonomic (i.e., classification) problem using multiple botanical measurements; analysis of variance and its big brother, multivariate analysis of variance, derived from a need to analyze data from agricultural experiments; and the origins of regression and correlation go back to problems involving heredity and the orbits of planets.

Each of these multivariate statistical techniques was created in an era when small or medium-sized data sets were common and, judged by today's standards, computing was carried out on less-than-adequate computational platforms (desk calculators, followed by mainframe batch computing with punched cards). Even as computational facilities improved dramatically (with the introduction of the minicomputer, the hand calculator, and the personal computer), it was only recently that the floodgates opened and the amounts of data recorded and stored began to surpass anything previously available. As a result, the focus of multivariate data analysis is changing rapidly, driven by a recognition that fast and efficient computation is of paramount importance to its future.

Statisticians have always been considered as partners for joint research in all the scientific disciplines. They are now beginning to participate with researchers from some of the subdisciplines within computer science, such as pattern recognition, neural networks, symbolic machine learning, computational learning theory, and artificial intelligence, and also with those working in the new field of bioinformatics; together, new tools are being devised for handling the massive quantities of data that are routinely collected in business transactions, governmental studies, science and medical research, and for making law and public policy decisions.

We are now seeing many innovative multivariate techniques being devised to solve large-scale data problems. These techniques include nonparametric density estimation, projection pursuit, neural networks, reduced-rank regression, nonlinear manifold learning, independent component analysis, kernel methods and support vector machines, decision trees, and random forests. Some of these techniques are new, but many of them are not so new (having been introduced several decades ago but virtually ignored by the statistical community). It is because of the current focus on large data sets that these techniques are now regarded as serious alternatives to (and, in some cases, improvements over) classical multivariate techniques.

This book focuses on the areas of regression, classification, and manifold learning, topics now regarded as the core components of data mining and machine learning, which we briefly describe in this chapter. It is important to note here that these areas overlap a great deal in content and methodology: what is one person's data-mining problem may be another's machine-learning problem.

1.2 Data Mining

1.2.1 From EDA to Data Mining

Although the revolutionary concept of *exploratory data analysis (EDA)* (Tukey, 1977) changed the way many statisticians viewed their discipline, emphasis in EDA centered on quick and dirty methods (using pencil and paper) for the visualization and examination of small data sets. Enthusiasts soon introduced EDA topics into university (and high school) courses in statistics. To complete the widespread acceptance and utility of John Tukey's exploratory procedures and his idiosyncratic nomenclature, EDA techniques were included in standard statistical software packages. Nevertheless, despite the available computational power, EDA was still perceived as a collection of small-sample, data-analytic tools.

Today, measurements on a variety of related variables often produce a data set so large as to be considered unwieldy for practical purposes. Such data now often range in size from moderate (say 10^3 to 10^4 cases) to large (10^6 cases or more). For example, billions of transactions each year are carried out by international finance companies; Internet traffic data are described as "ferocious" (Cleveland and Sun, 2000); the Human Genome Project has to deal with gigabytes (2^{30} ($\sim 10^9$) bytes) of genetic information; astronomy, the space sciences, and the earth sciences have terabytes (2^{40} ($\sim 10^{12}$) bytes) and soon, petabytes (2^{50} ($\sim 10^{15}$) bytes), of data for processing; and remote-sensing satellite systems, in general, record many gigabytes of data each hour. Each of these data sets is incredibly large and complex, with millions of observations being recorded on huge numbers of variables.

Furthermore, governmental statistical agencies (e.g., the Federal Statistical Service in the United States, the National Statistical Service in the United Kingdom, and similar agencies in other countries) are accumulating greater amounts of detailed economic, labor, demographic, and census information than at any time in the past. The U.S. census file based solely on administrative records, for example, has been estimated to be of size at least 10^{12} bytes (Kirkendall, 1997). Other massive data sets (e.g., crime data, health-care data) are maintained by other governmental agencies.

The availability of massive quantities of data coupled with enormous increases in computational power for relatively low cost has led to the creation of a whole new activity called *data mining*. With massive data sets, the process of data mining is not unlike a gigantic effort at EDA for "infinite" data sets. For many companies, their data sets of interest are so large that only the simplest of statistical computations can be carried out. In such situations, data mining means little more than computing means and standard deviations of each variable; drawing some bivariate scatterplots and carrying out simple linear regressions of pairs of variables; and doing some cross-tabulations. The level of sophistication of a data mining study depends not just on the statistical software but also on the computer hardware (RAM, hard disk, etc.) and database management system for storing the data and processing the results.

Even if we are faced with a huge amount of data, if the problem is simple enough, we can sample and use standard exploratory and confirmatory methods. In some instances, especially when dealing with government-collected data, sampling may be carried out by the agency itself. Census data, for example, is too big to be useful for most users; so, the U.S. Census Bureau creates manageable public-use files by drawing a random sample of individuals from the full data set and either removes or masks identifying information (Kirkendall, 1997),

In most applications of data mining, there is no à priori reason to sample. The entire population of data values (at least, those with which we would be interested) is readily available, and the questions asked of that data set are usually exploratory in nature and do not involve inference. Because a data pattern (e.g., outliers, data errors, hidden trends, credit-card fraud) is a local phenomenon, possibly affecting only a few observations, sampling, which typically reduces the size of the data set in drastic fashion, may completely miss the specifics of whatever pattern would be of special interest.

Data mining differs from classical statistical analysis in that statistical inference in its hypothesis-testing sense may not be appropriate. Furthermore, most of the questions asked of large data sets are different from the classical inference questions asked of much smaller samples of data. This is not to say that sampling and subsequent modeling and inference have no role to play when dealing with massive data sets. Sampling, in fact, may be appropriate in certain circumstances as an accompaniment to any detailed data exploration activities.

1.2.2 What Is Data Mining?

It is usual to categorize data mining activities as either *descriptive* or *predictive*, depending upon the primary objective:

Descriptive data mining: Search massive data sets and discover the locations of unexpected structures or relationships, patterns, trends, clusters, and outliers in the data.

Predictive data mining: Build models and procedures for regression, classification, pattern recognition, or machine learning tasks, and assess the predictive accuracy of those models and procedures when applied to fresh data.

The mechanism used to search for patterns or structure in high-dimensional data might be manual or automated; searching might require interactively querying a database management system, or it might entail using visualization software to spot anomolies in the data. In machine-learning terms, descriptive data mining is known as *unsupervised learning*, whereas predictive data mining is known as *supervised learning*.

Most of the methods used in data mining are related to methods developed in statistics and machine learning. Foremost among those methods are the general topics of regression, classification, clustering, and visualization. Because of the enormous sizes of the data sets, many applications of data mining focus on dimensionality-reduction techniques (e.g., variable selection) and situations in which high-dimensional data are suspected of lying on lower-dimensional hyperplanes. Recent attention has been directed to methods of identifying high-dimensional data lying on nonlinear surfaces or manifolds.

Table 1.1 lists some of the application areas of data mining and examples of major research themes within those areas. Using the massive data sets that are routinely collected by each of these disciplines, advances in dealing with the topics depend crucially upon the availability of effective data mining techniques and software.

One of the most important issues in data mining is the computational problem of *scalability*. Algorithms developed for computing standard exploratory and confirmatory statistical methods were designed to be fast and computationally efficient when applied to small and medium-sized data sets; yet, it has been shown that most of these algorithms are not up to the challenge of handling huge data sets. As data sets grow, many existing algorithms demonstrate a tendency to slow down dramatically (or even grind to a halt).

In data mining, regardless of size or complexity of the problem (essentially, the numbers of variables and observations), we require algorithms to have good performance characteristics; that is, they have to be scalable. There is no globally accepted definition of scalability, but a general idea of what this property means is the following:

Scalability: The capability of an algorithm to remain efficient and accurate as we increase the complexity of the problem.

The best scenario is that scalability should be linear. So, one goal of data mining is to create a library of scalable algorithms for the statistical analysis of large data sets.

Another issue that has to be considered by those working in data mining is the thorny problem of *statistical inference*. The twentieth century saw Fisher, Neyman, Pearson, Wald, Savage, de Finetti, and others provide a variety of competing — yet related — mathematical frameworks (frequentist, Bayesian, fiducial, decision theoretic, etc.) from which inferential theories of statistics were built. Extrapolating to a future point in time, can we expect researchers to provide a version of statistical inference for analyzing massive data sets?

There are situations in data mining when statistical inference — in its classical sense — either has no meaning or is of dubious validity: the former occurs when we have the entire population to search for answers (e.g., gene or protein sequences, astronomical recordings), and the latter occurs when a data set is a "convenience" sample rather than being a random sample drawn from some large population. When data are collected through time (e.g., retail transactions, stock-market transactions, patient records, weather records), sampling also may not make sense; the time-ordering of the observations is crucial to understanding the phenomenon generating the data, and to treat the observations as independent when they may be highly correlated will provide biased results.

Those who now work in data mining recognize that the central components of data mining are — in addition to statistical theory and methods — computing and computational efficiency, automatic data processing, dynamic and interactive data visualization techniques, and algorithm development. There are a number of software packages whose primary purpose is to help users carry out various techniques in data mining. The leading data-mining products include the packages listed (in alphabetical order) in Table 1.2.

TABLE 1.1. *Application areas of data mining*

Marketing: Predict new purchasing trends. Identify “loyal” customers. Predict what types of customers will respond to direct mailings, telemarketing calls, advertising campaigns, or promotions. Given customers who have purchased product A, B, or C, identify those who are likely to purchase product D and, in general, which products sell together (popularly called *market basket analysis*).

Banking: Predict which customers will likely switch from one credit card company to another. Evaluate loan policies using customer characteristics. Predict behavioral use of automated teller machines (ATMs).

Financial Markets: Identify relationships between financial indicators. Track changes in an investment portfolio and predict price turning points. Analyze volatility patterns in high-frequency stock transactions using volume, price, and time of each transaction.

Insurance: Identify characteristics of buyers of new policies. Find unusual claim patterns. Identify “risky” customers.

Healthcare: Identify successful medical treatments and procedures by examining insurance claims and billing data. Identify people “at risk” for certain illnesses so that treatment can be started before the condition becomes serious. Predict doctor visits from patient characteristics. Use healthcare data to help employers choose between HMOs.

Molecular Biology: Collect, organize, and integrate the enormous quantities of data on bioinformatics, functional genomics, proteomics, gene expression monitoring, and microarrays. Analyze amino acid sequences and deoxyribonucleic acid (DNA) microarrays. Use gene expression to characterize biological function. Predict protein structure and identify related proteins.

Astronomy: Catalogue (as stars, galaxies, etc.) hundreds of millions of objects in the sky using hundreds of attributes, such as position, size, shape, age, brightness, and color. Identify patterns and relationships of objects in the sky.

Forensic Accounting: Identify fraudulent behavior in credit card usage by looking for transactions that do not fit a particular cardholder’s buying habits. Identify fraud in insurance and medical claims. Identify instances of tax evasion. Detect illegal activities that can lead to suspected money laundering operations. Identify stock market behaviors that indicate possible insider-trading operations.

Sports: Identify in realtime which players and which designed plays are most effective at specific points in the game and in relation to combinations of opposing players. Identify the exact moment when intriguing play patterns occurred. Discover game patterns hidden behind summary statistics.

TABLE 1.2. *Data mining software packages.*

Company	Software Package
IBM Corp.	*Intelligent Miner*
Insightful	*Insightful Miner*
NCR Corp.	*Teradata Warehouse Miner*
Oracle	*Darwin*
SAS Institute, Inc.	*Enterprise Miner*
Silicon Graphics, Inc.	*MineSet*
SPSS, Inc.	*Clementine*

1.2.3 Knowledge Discovery

Data mining has been described (Fayyad, Piatetsky-Shapiro, and Smyth, 1996) as a step in a more general process known as *knowledge discovery in databases (KDD)*. The "knowledge" acquired by KDD has to be interesting, non-trivial, non-obvious, previously unknown, and potentially useful.

KDD is a multistep process designed to assist those who need to search huge data sets for "nuggets of useful information." In KDD, assistance is expected to be intelligent and automated, and the process itself is interactive and iterative.

KDD is composed of six primary activities:

1. selecting the target data set (which data set or which variables and cases are to be used for data mining);

2. data cleaning (removal of noise, identification of potential outliers, imputing missing data);

3. preprocessing the data (deciding upon data transformations, tracking time-dependent information);

4. deciding which data-mining tasks are appropriate (regression, classification, clustering, etc.);

5. analyzing the cleaned data using data-mining software (algorithms for data reduction, dimensionality reduction, fitting models, prediction, extracting patterns);

6. interpreting and assessing the knowledge derived from data-mining results.

In KDD, and hence in data mining, the descriptive aspect is more important than the predictive aspect, which forms the main goal of machine learning.

1.3 Machine Learning

Machine learning evolved out of the subfield of computer science known as *artificial intelligence (AI)*. Whereas the focus of AI is to make machines intelligent, able to think rationally like humans and solve problems, machine learning is concerned with creating computer systems and algorithms so that machines can "learn" from previous experience. Because intelligence cannot be attained without the ability to learn, machine learning now plays a dominant role in AI.

1.3.1 How Does a Machine Learn?

A machine learns when it is able to accumulate experience (through data, programs, etc.) and develop new knowledge so that its performance on specific tasks improves over time. This idea of learning from experience is central to the various types of problems encountered in machine learning, especially problems involving classification (e.g., handwritten digit recognition, speech recognition, face recognition, text classification). The general goal of each of these problems is to find a systematic way of classifying a future example (e.g., a handwriting sample, a spoken word, a face image, a text fragment). Classification is based upon measurements on that future example together with knowledge obtained from a *learning* (or *training*) *sample* of similar examples (where the class of each example is completely determined and known, and the number of classes is finite and known).

The need to create new methods and terminology for analyzing large and complex data sets has led to researchers from several disciplines — statistics, pattern recognition, neural networks, symbolic machine learning, computational learning theory, and, of course, AI — to work together to influence the development of machine learning.

Among the techniques that have been used to solve machine-learning problems, the topics that are of most interest to statisticians — density estimation, regression, and pattern recognition (including neural networks, discriminant analysis, tree-based classifiers, random forests, bagging and boosting, support vector machines, clustering, and dimensionality-reduction methods) — are now collectively referred to as *statistical learning* and constitute many of the topics discussed in this book. Vladimir N. Vapnik, one of the founders of statistical learning theory, relates statistics to learning theory in the following way (Vapnik, 2000, p. x):

> *The problem of learning is so general that almost any question that has been discussed in statistical science has its analog in learning theory. Furthermore, some very important general results were first found in the framework of learning theory and then formulated in the terms of statistics.*

The machine-learning community divides learning problems into various categories: the two most relevant to statistics are those of *supervised learning* and *unsupervised learning.*

Supervised learning: Problems in which the learning algorithm receives a set of continuous or categorical input variables and a correct output variable (which is observed or provided by an explicit "teacher") and tries to find a function of the input variables to approximate the known output variable: a continuous output variable yields a regression problem, whereas a categorical output variable yields a classification problem.

Unsupervised learning: Problems in which there is no information available (i.e., no explicit "teacher") to define an appropriate output variable; often referred to as "scientific discovery."

The goal in unsupervised learning differs from that of supervised learning. In supervised learning, we study relationships between the input and output variables; in unsupervised learning, we explore particular characteristics of the input variables only, such as estimating the joint probability density, searching out clusters, drawing proximity maps, locating outliers, or imputing missing data.

Sometimes there might not be a "bright-line" distinction between supervised and unsupervised learning. For example, the dimensionality-reduction technique of principal component analysis (PCA) has no explicit output variable and, thus, appears to be an unsupervised-learning method; however, as we will see, PCA can be formulated in terms of a multivariate regression model where the input variables are also used as output variables, and so PCA can also be regarded as a supervised-learning method.

1.3.2 Prediction Accuracy

One of the most important tasks in statistics is to assess the accuracy of a predictor (e.g., regression estimator or classifier). The measure of prediction accuracy typically used is that of *prediction error*, defined generically as

Prediction error: In a regression problem, the mean of the squared errors of prediction, where error is the difference between a true output value and its corresponding predicted output value; in a classification problem, the probability of misclassifying a case.

The simplest estimate of *prediction error* is the *resubstitution error*, which is computed as follows. In a regression problem, the fitted model is used to predict each of the (known) output values from the entire data set, and the resubstitution estimate is then the mean of the squared residuals, also known as the *residual mean square*. In a classification problem, the classifier predicts the (known) class of each case in the entire data set, a correct prediction is scored as a 0 and a misclassification is scored as a 1, and the resubstitution estimate is the proportion of misclassified cases.

Because the resubstitution estimate uses the same data as was used to derive the predictor, the result is an overly optimistic view of prediction accuracy. Clearly, it is important to do better.

1.3.3 Generalization

The need to improve upon the resubstitution estimator of prediction accuracy led naturally to the concept of *generalization*: we want an estimation procedure to generalize well; that is, to make good predictions when applied to a data set *independent of that used to fit the model*. Although this is not a new idea — it has existed in statistics for a long time (see, e.g., Mosteller and Tukey, 1977, pp. 37–38) — the machine-learning community embraced this particular concept (adopting the name from psychology) and made it a central issue in the theory and applications of machine learning.

Where do we find such an independent data set? One way is to gather fresh data. However, "when fresh gathering is not feasible, good results can come from going to a body of data that has been kept in a locked safe where it has rested untouched and unscanned during all the choices and optimizations" (Mosteller and Tukey, 1977, p. 38). The data in the "locked safe" can be viewed as holding back a portion of the current data from the model-fitting phase and using it instead for assessment purposes. If an independent set of data is not used, then we will overestimate the model's predictive accuracy.

In fact, it is now common practice — assuming the data set is large enough — to use a random mechanism to separate the data into three nonoverlapping and independent data sets:

a learning (or training) set $\mathcal{L}$, a data set where "anything goes ... including hunches, preliminary testing, looking for patterns, trying large numbers of different models, and eliminating outliers" (Efron, 1982, p. 49);

a validation set $\mathcal{V}$, a data set to be used for model selection and assessment of competing models (usually on the basis of predictive ability);

a test set $\mathcal{T}$, a data set to be used for assessing the performance of a completely specified final model.

The key assumption here is that the three subsets of the data are each generated by the same underlying distribution. In some instances, learning data may be taken from historical records.

As a simple guideline, the learning set should consist of about 50% of the data, whereas the validation and test sets may each consist of 25% (although these percentages are not written in stone). In some instances, we may find it convenient to merge the validation set with the test set, thus forming a larger test set. For example, we often see publicly available data sets in Internet databases divided into a learning set and a test set.

1.3.4 Generalization Error

In supervised learning problems, it is important to assess how closely a particular model (function of the inputs) fits the data (the outputs). As before, we use prediction error as our measure of prediction accuracy.

In regression problems, there are two different types of prediction error. For both types, we first fit a model to the learning set $\mathcal{L}$. Then, we use that fitted model to predict the output values of either $\mathcal{L}$ (given input values from $\mathcal{L}$) or the test set $\mathcal{T}$ (given input values from $\mathcal{T}$). Prediction error is the mean (computed only over the appropriate data set) of the squared-errors of prediction (where error = true output value − predicted output value). If we average over $\mathcal{L}$, the prediction error is called the *regression learning error* (equivalent to the resubstitution estimate computed only over $\mathcal{L}$), whereas if we average over $\mathcal{T}$, the prediction error is called the *regression test error*.

A similar strategy is used in classification problems; only the definition of prediction error is different. We first build a classifier from $\mathcal{L}$. Next, we use that classifier to predict the class of each data vector in either $\mathcal{L}$ or $\mathcal{T}$. For each prediction, we assign the value of 0 to a correct classification and 1 to a classification error. The prediction error is then defined as the average of all the 0s and 1s over the appropriate data set (i.e., the proportion of misclassified observations). If we average over $\mathcal{L}$, then prediction error is referred to as the *classification learning error* (equivalent to the resubstitution estimate computed only over $\mathcal{L}$), whereas averaging over $\mathcal{T}$ yields the *classification test error*.

If the learning set $\mathcal{L}$ is moderately sized, we may feel that using only a portion of the entire data set to fit the model is a waste of good data. Alternative data-splitting methods for estimating test error are based upon *cross-validation* (Stone, 1974) and the *bootstrap* (Efron, 1979):

V-fold cross-validation: Randomly divide the entire data set into, say, V nonoverlapping groups of roughly equal size; remove one of the groups and fit the model using the combined data from the other $V-1$ groups (which forms the learning set); use the omitted group as the test set, predict its output values using the fitted model, and compute the prediction error for the omitted group; repeat this procedure V times, each time removing a different group; then, average the resulting V

prediction errors to estimate the test error. The number of groups V can be any number from 2 to the sample size.

Bootstrap: Select a "bootstrap sample" from the entire data set by drawing a random sample *with replacement* having the same size as the parent data set, so that the sample may contain repeated observations; fit a model using this bootstrap sample and compute its prediction error; repeat this sampling procedure, say, 1000 times, each time computing a prediction error; then, average all the prediction errors to estimate the test error.

These are generic descriptions of the two procedures; specific descriptions are given in various sections of this book. In particular, the definition of the bootstrap is actually more complicated than that given by this description because it depends on what is assumed about the stochastic model generating the data. Although both cross-validation and the bootstrap are computationally intensive techniques, cross-validation uses the entire data set in a more efficient manner than the division into a learning set and an independent test set. We also caution that, in some applications, it may not make sense to use one of these procedures.

The expected prediction error over an independent test set is called *infinite test error* or *generalization error*. We estimate generalization error by the test error. One goal of *generalization theory* is to choose that regression model or classifier that gives the smallest generalization error.

1.3.5 Overfitting

To minimize generalization error, it is tempting to find a model that will fit the data in the learning set as accurately as possible. This is not usually advisable because it may make the selected model too complicated. The resulting learning error will be very small (because the fitted model has been optimized for that data set), whereas the test error will be large (a consequence of *overfitting*).

Overfitting: Occurs when the model is too large or complicated, or contains too many parameters relative to the size of the learning set. It usually results in a very small learning error and a large generalization (test) error.

One can control such temptation by following the principle known as *Ockham's razor*, which encourages us to choose simple models while not losing track of the need for accuracy. Simple models are generally preferred if either the learning set is too small to derive a useful estimate of the model or fitting a more complex model would necessitate using huge amounts of computational resources.

We illustrate the idea of overfitting with a simple regression example. Using 10 equally spaced x values as the learning set, we generate corresponding y values from the function $y = 0.5 + 0.25\cos(2\pi x) + e$, where the Gaussian noise component e has mean zero and standard deviation 0.06. We try to approximate the underlying unknown function (the cosinusoid) by a polynomial in x, where the problem is to decide on the degree of the polynomial. In the top-left panel of Figure 1.1, we give the cosinusoid and the 10 generated points; in the top-right panel, a linear regression function gives a poor fit to the points and shows the result of *underfitting* by using too few parameters; in the bottom-left panel, a cubic polynomial is fitted to the data, showing an improved approximation to the cosinusoid; and in the bottom-right panel, by increasing the fit to a $9th$-degree polynomial, we ensure that the fitted curve passes through each point exactly. However, the $9th$-degree polynomial actually makes the fit much worse by introducing unwanted fluctuations and shows the result of overfitting by using too many parameters.

How would such polynomial fits affect a test set obtained by using the same x values but different noise values (hence, different y values) in the above cosinusoid model? In Figure 1.2, we plot the prediction errors for both the learning set and the test set. The learning error, as expected, decreases monotonically to zero when we fit a $9th$-degree polynomial. This behavior for the learning error is typical whenever the fitted model ranges from the very simple to the most complex. The test error decreases to a $4th$ degree polynomial and then increases, indicating that models with too many parameters will have poor generalization properties.

Researchers have suggested several methods for reducing the effects of overfitting. These include methods that employ some form of averaging of predictions made by a number of different models fit to the learning set (e.g., the "bagging" and "boosting" algorithms of Chapter 14) and regularization (where complex models are penalized in favor of simpler models). Bayesian arguments in favor of a related idea of "model averaging" have also been proposed (see Hoeting, Madigan, Raftery, and Volinsky, 1999, for an excellent review of the topic).

1.4 Overview of Chapters

This book is divided into 17 chapters. Chapter 2 describes multivariate data, database management systems, and data problems. Chapter 3 reviews basic vector and matrix notation, introduces random vectors and matrices and their distributions, and derives maximum likelihood estimates for the multivariate Gaussian mean, including the James–Stein shrinkage estimator. Chapter 4 provides the elements of nonparametric density estimation. Chapters 5 reviews topics in multiple linear regression, including

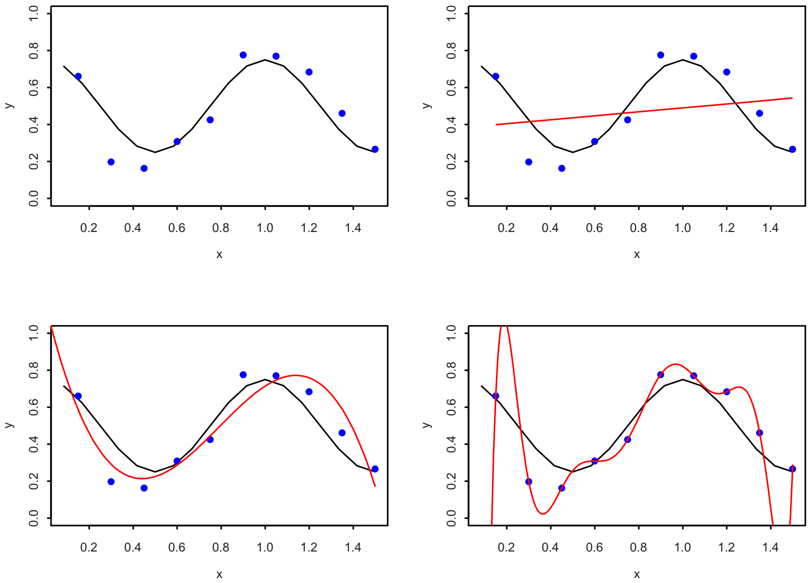

FIGURE 1.1. *Ten* y*-values corresponding to equally spaced* x*-values were generated from the cosinusoid* $y = 0.5 + 0.25\cos(2\pi x) + e$*, where the noise component* $e \sim \mathcal{N}(0, (0.06)^2)$*. Top-left panel: the true cosinusoid is shown in black with the 10 points in blue; top-right: the red line is the ordinary least-squares (OLS) linear regression fit to the points; bottom-left: the red curve is an OLS cubic polynomial fit to the points; bottom-right: the red curve is a 9th-degree polynomial that passes through every point.*

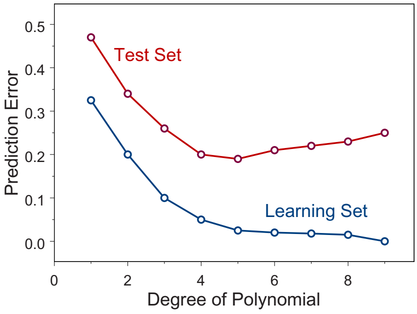

FIGURE 1.2. *Prediction error from the learning set (blue curve) and test set (red curve) based upon polynomial fits to data generated from a cosinusoid curve with noise.*

model assessment (through cross-validation and the bootstrap), biased regression, shrinkage, and model selection, concepts that will be needed in later chapters.

In Chapter 6, we discuss multivariate regression for both the fixed-X and random-X cases. We discuss multivariate analysis of variance and multivariate reduced-rank regression (RRR). RRR provides the foundation for a unified theory of multivariate analysis, which includes as special cases the classical techniques of principal component analysis, canonical variate analysis, linear discriminant analysis, factor analysis, and correspondence analysis. In Chapter 7, we introduce the idea of (linear) dimensionality reduction, which includes principal component analysis, canonical variate and correlation analysis, and projection pursuit. Chapter 8 discusses Fisher's linear discriminant analysis. Chapter 9 introduces recursive partitioning and classification and regression trees. Chapter 10 discusses artificial neural networks via analogies to neural networks in the brain, artificial intelligence, and expert systems, as well as the related statistical techniques of projection pursuit regression and generalized additive models. Chapter 11 deals with classification using support vector machines. Chapter 12 describes the many algorithms for cluster analysis and unsupervised learning.

In Chapter 13, we discuss multidimensional scaling and distance geometry, and Chapter 14 introduces committee machines and ensemble methods, such as bagging, boosting, and random forests. Chapter 15 discusses independent component analysis. Chapter 16 looks at nonlinear methods for dimensionality reduction, especially the various flavors of nonlinear principal component analysis, and nonlinear manifold learning. Chapter 17 describes correspondence analysis.

Bibliographical Notes

Books on data mining include Fayyad, Piatetsky-Shapiro, Smyth, and Uthurusamy (1996) and Hand, Mannila, and Smyth (2001). There are annual KDD workshops and conferences and a KDD journal. There is a KDD section of the ACM: `www.acm.org/sigkdd`. Books on machine learning include Bishop (1995), Ripley (1996), Hastie, Tibshirani, and Friedman (2001), MacKay (2003), and Bishop (2006).