Introduction

Multivariate Analysis

This course 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

1. 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.

2. 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.

3. 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.

Advantages of Multivariate Analysis

Multivariate statistical analysis is the simultaneous statistical analysis of a collection of random variables.

Multivariate analysis improves separate univariate analyses of each variable in a study because it incorporates information into the statistical analysis about the relationships between all the variables.

Early Developments

Much of the early developmental work in multivariate analysis was motivated by problems from the social and behavioral sciences.

- 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.

- Multidimensional scaling originated in psychometrics, where it was used to understand people's judgments of the similarity of items in a set.

- 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.

- 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.

From EDA to Data Mining

Although the revolutionary concept of exploratory data analysis (EDA) 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.

Big Data

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).

- Billions of transactions each year are carried out by international finance companies.
- Internet traffic data are described as ferocious.

- 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.

- 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.

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 scatter plots 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.

In most applications of data mining, there is no a 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.

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.

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.

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).

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.

Knowledge Discovery

Data mining has been described 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.

Primary Activities of KDD

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.

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.

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 course.

Supervised 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, 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.

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.

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.

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 – 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.

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

1. Randomly divide the entire data set into, say, V (can be any number from 2 to the sample size) nonoverlapping groups of roughly equal size;

2. remove one of the groups and fit the model using the combined data from the other $V-1$ groups (which forms the learning set);

3. 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;

4. repeat this procedure V times, each time removing a different group;

5. average the resulting V prediction errors to estimate the test error.

Bootstrap

1. 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;

2. fit a model using this bootstrap sample and compute its prediction error;

3. repeat this sampling procedure, say, 1000 times, each time computing a prediction error;

4. 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 course.

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.

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.

Ockham's Razor

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.

Researchers have suggested several methods for reducing the effects of over-fitting.

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) 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.

Data Sets

Multivariate data consist of multiple measurements, observations, or responses obtained on a collection of selected variables.

The types of variables usually encountered often depend upon those who collect the data (the domain experts), possibly together with some statistical colleagues; for it is these people who actively decide which variables are of interest in studying a particular phenomenon.

In other circumstances, data are collected automatically and routinely without a research direction in mind, using software that records every observation or transaction made regardless of whether it may be important or not.

Data are raw facts, which can be numerical values (e.g., age, height, weight), text strings (e.g., a name), curves (e.g., a longitudinal record regarded as a single functional entity), or two-dimensional images (e.g., photograph, map).

When data sets are small in size, we find it convenient to store them in spreadsheets or as flat files (large rectangular arrays). We can then use any statistical software package to import such data for subsequent data analysis, graphics, and inference.

As mentioned in previous sections, massive data sets are now sprouting up everywhere. Data of such size need to be stored and manipulated in special database systems.

Data Examples

- DNA Microarray Data
- Mixtures of Polyaromatic Hydrocarbons
- Face Recognition
- ...

Databases

A database is a collection of persistent data, where by persistent we mean data that can be removed from the database only by an explicit request and not through an application's side effect.

The most popular format for organizing data in a database is in the form of tables (also called data arrays or data matrices), each table having the form of a rectangular array arranged into rows and columns, where a row represents the values of all variables on a single multivariate observation (response, case, or record), and a column represents the values of a single variable for each observation.

In this course, a typical database table having n multivariate observations taken on r variables will be represented by an $(r \times n)$-matrix,

$$\mathcal{X}_{r\times n} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & x_{r2} & \cdots & x_{rn} \end{pmatrix}.$$

Although database tables are set up to have the form of $\mathcal{X}^T$, with variables as columns and observations as rows, we will find it convenient to set $\mathcal{X}$ to be the transpose of the database table.

Data Types

- Indexing
- Binary/Boolean
- Categorical: Nominal and Ordinal
- Integer/Count
- Continuous

Fixed and Stochastic

- Fixed: The values of a fixed variable have deliberately been set in advance, as in a designed experiment, or are considered causal to the phenomenon in question; as a result, interest centers only on a specific group of responses. This category usually refers to indexing variables but can also include some of the above types.

- Stochastic: The values of a stochastic variable can be considered as having been chosen at random from a potential list (possibly, the real line or a portion of it) in some stochastic manner. In this sense, the values obtained are representative of the entire range of possible values of the variable in question.

Input and Output Variables

- Input variable: Also called a predictor or independent variable, typically denoted by X, and may be considered to be fixed (or preset or controlled) through a statistically designed experiment, or stochastic if it can take on values that are observed but not controlled.

- Output variable: Also called a response or dependent variable, typically denoted by Y, and which is stochastic and dependent upon the input variables.

Databases on the Internet

Table: Internet websites containing many different databases.

www.ics.uci.edu/pub/machine-learning-databases
lib.stat.cmu.edu/datasets
www.statsci.org/datasets.html
www.amstat.org/publications/jse/jsedataarchive.html
www.physionet.org/physiobank/database

TABLE: Internet websites containing microarray databases.

biostat.mc.vanderbilt.edu/twiki/bin/view/Main/DataSets
www.broad.mit.edu/tools/data.html
sdmc.lit.org.sg/GEDatasets/Datasets.html
genome-www5.stanford.edu
www.bioconductor.org/packages/1.8/AnnotationData.html
www.ncbi.nlm.nih.gov/geo

TABLE: Internet websites containing natural-language text databases.

arXiv.org
medir.ohsu.edu/pub/ohsumed
kdd.ics.uci.edu/databases/reuters21578/reuters21578.html
kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html

Database Management

- Elements of Database Systems
- Structured Query Language (SQL)
- OLTP Databases
- Integrating Distributed Databases
- Data Warehousing
- Decision Support Systems and OLAP

R Packages and DBMSs

The R package RODBC (written by Michael Lapsley and Brian Ripley, and available from CRAN) provides an R interface to DBMSs based upon the Microsoft ODBC (Open Database Connectivity) standard.

RODBC, which runs on both MS Windows and Unix/Linux, is able to copy an R data frame to a table in a database (command: sqlSave), read a table from a DBMS into an R data frame (sqlFetch), submit an SQL query to an ODBC database (sqlQuery), retrieve the results (sqlGetResults), and update the table where the rows already exist (sqlUpdate).

RODBC works with Oracle, MS Access, Sybase, DB2, MySQL, PostgreSQL, and SQL Server on MS Windows platforms and with MySQL, PostgreSQL, and Oracle under Unix/Linux.

Data Quality Problems

Errors exist in all kinds of databases.

Those that are easy to detect will most likely be found at the data cleaning stage, whereas those errors that can be quite resistant to detection might only be discovered during data analysis.

Data cleaning usually takes place as the data are received and before they are stored in read-only format in a data warehouse.

A consistent and cleaned-up version of the data can then be made available.

Data Inconsistencies

Errors in compiling and editing the resulting database are common and actually occur with alarming frequency, especially in cases where the data set is very large.

When data from different sources are being connected, inconsistencies as to a person's name (especially in cases where a name can be spelled in several different ways) occur frequently, and matching (or disambiguation) has to take place before such records can be merged.

One popular solution is to employ Soundex (sound-indexing) techniques for name matching.

Massive data sets are prone to mistakes, errors, distortions, and, in general, poor data quality, just as is any data set, but such defects occur here on a far grander scale because of the size of the data set itself.

When invalid product codes are entered for a product, they may easily be detected; when valid product codes, however, are entered for the wrong product, detection becomes more difficult.

Outliers

Outliers are values in the data that, for one reason or another, do not appear to fit the pattern of the other data values; visually, they are located far away from the rest of the data.

It is not unusual for outliers to be present in a data set.

Outliers can occur for many different reasons but should not be confused with gross errors.

Gross errors are cases where something went wrong; they include human errors (e.g., a numerical value recorded incorrectly) and mechanical errors (e.g., malfunctioning of a measuring instrument or a laboratory instrument during analysis).

The density of gross errors depends upon the context and the quality of the data. In medical studies, gross error rates in excess of 10% have been quoted.

Univariate outliers are easy to detect when they indicate impossible (or out of bounds) values.

More often, an outlier will be a value that is extreme, either too large or too small.

For multivariate data, outlier detection is more difficult. Low-dimensional visual displays of the data (such as histograms, boxplots, scatterplots) can encourage insight into the data and provide at the same time a method for manually detecting some of the more obvious univariate or bivariate outliers.

When we have a large data set, outliers may not be all that rare.

Unlike a data set of 100 or so observations, where we may find two or three outliers, in a data set of 100,000, we should not be surprised to discover a large number (in some cases, hundreds, and maybe even thousands) of outliers.

To detect true multidimensional outliers, however, becomes a test of statistical ingenuity. A multivariate observation whose every component value may appear indistinguishable from the rest may yet be regarded as an outlier when all components are treated simultaneously.

In large multivariate data sets, some combination of visual display of the data, manual outlier detection scheme, and automatic outlier detection program may be necessary: potential outliers could be flagged by an automatic screening device, and then an analyst would manually decide on the fate of that flagged outlier.

Missing Data

In the vast majority of data sets, there will be missing data values.

For example, human subjects may refuse to answer certain items in a battery of questions because personal information is requested; some observations may be accidentally lost; some responses may be regarded as implausible and rejected; and in a study of financial records of a company, some records may not be available because of changes in reporting requirements and data from merged or reorganized organizations.

One popular method deletes those observations that contain missing data and analyzes only those cases that are observed in their entirety (often called complete-case analysis or listwise-deletion method).

Such a complete-case analysis may be satisfactory if the proportion of deleted observations is small relative to the size of the entire data set and if the mechanism that leads to the missing data is independent of the variables in question.

If the missing data constitute a sizeable proportion of the entire data set, then complete-case methods will not work.

Single imputation

Single imputation has been used to impute (or fill in) an estimated value for each missing observation and then analyze the amended data set as if there had been no missing values in the first place.

- **hot-deck imputation:** a missing value is imputed by substituting a value from a similar but complete record in the same data set;
- **mean imputation:** the singly imputed value is just the mean of all the completely recorded values for that variable;
- **regression imputation**: uses the value predicted by a regression on the completely recorded data.

Note

Because sampling variability due to single imputation cannot be incorporated into the analysis as an additional source of variation, the standard errors of model estimates tend to be underestimated.

More Variables than Observations

Many statistical computer packages do not allow the number of input variables, r, to exceed the number of observations, n, because, then, certain matrices, such as the $(r \times r)$ covariance matrix, would have less than full rank, would be singular, and, hence, uninvertible.

Yet, we should not be surprised when $r > n$.

Typical Examples

- Satellite images
- Chemometrics
- Gene expression data

When $r > n$, one way of dealing with this problem is to analyze the data on each variable separately.

However, this suggestion does not take account of correlations between the variables.

Researchers have recently provided new statistical techniques that are not sensitive to the $r > n$ issue.

We will address this situation in various sections of this course.

The Curse of Dimensionality

The term **curse of dimensionality** (Bellman, 1961) originally described how difficult it was to perform high-dimensional numerical integration.

This led to the more general use of the term to describe the difficulty of dealing with statistical problems in high dimensions.

Implications

1. We can never have enough data to cover every part of high-dimensional input space to learn which part of the space is important to a relationship and which is not.

2. As the number of dimensions grows larger, almost all the volume inside a hypercubic region of input space lies closer to the boundary or surface of the hypercube rather than near the center.