Chapter 1

DATA MINING AT THE INTERFACE OF COMPUTER SCIENCE AND STATISTICS *

Padhraic Smyth
Information and Computer Science
University of California, Irvine
CA 92697–3425
smyth@ics.uci.edu

Abstract This chapter is written for computer scientists, engineers, mathematicians, and scientists who wish to gain a better understanding of the role of statistical thinking in modern data mining. Data mining has attracted considerable attention both in the research and commercial arenas in recent years, involving the application of a variety of techniques from both computer science and statistics. The chapter discusses how computer scientists and statisticians approach data from different but complementary viewpoints and highlights the fundamental differences between statistical and computational views of data mining. In doing so we review the historical importance of statistical contributions to machine learning and data mining, including neural networks, graphical models, and flexible predictive modeling. The primary conclusion is that closer integration of computational methods with statistical thinking is likely to become increasingly important in data mining applications.

Keywords: Data mining, statistics, pattern recognition, transaction data, correlation.

1. Introduction

The goal of this chapter is to explore the past, present, and potential future relationship of statistics to data mining, and to further argue that

*Invited chapter, *Data Mining for Scientific and Engineering Applications*, to appear 2001.

statistics should play a foundational role in any data mining endeavor. The target audience of this chapter is intended to be computer scientists, engineers, and scientists who are knowledgeable about the algorithmic aspects of data mining but who would like to learn more about how statistical ideas and concepts may help them in their work. Hopefully this chapter will provide at least a few pointers for such readers.

The term "data mining" has been used in a variety of contexts in data analysis in recent years. As a reader of this book it is likely that you already think of data mining from the computer science viewpoint, namely, as a broad set of techniques and algorithms for extracting useful patterns and models from very large data sets. Since the early 1990's there has been a broad surge of both research and commmercial activity in this area, largely driven by computationally-efficient massive search for patterns in data (such as association rules) and by business applications such as analysis of very large transactional data archives. As evidenced by well-known business-oriented data mining texts (e.g., [BL00]), as well as various published proceedings of data mining research conferences, much current work in data mining is focused on algorithmic issues such as computational efficiency and on data engineering issues such as data representation. While these are important topics in their own right, statistical considerations and analyses are often conspicuous by their absence in data mining texts and papers. This is a natural consequence of the fact that research in data mining is largely practiced by computer scientists who naturally focus on algorithmic and computational issues rather than on statistical issues. In this chapter we explore the interplay of computer science and statistics and how this has impacted, and will impact, the development of data mining.

2. Is Data Mining Different from Statistics?

Is data mining as currently practiced substantially different from conventional applied statistics? Certainly if one looks at the published commercial applications of data mining, such as the case studies presented in [BL00], one sees a heavy reliance on techniques that have their lineage in applied statistics. For example, decision trees are perhaps the single most widely-used modeling technique in commercial predictive data mining applications [Joh99, Koh00]. They are particularly popular because of their ability to both deal with heterogenous data types (they can easily handle both categorical and real-valued variables) and to find relatively low-dimensional parsimonious predictors for high-dimensional problems. Many other techniques that are popular in data mining also have their roots in applied statistics, such as nearest neighbor models, naive Bayes models, and logistic regression for prediction, and k-means and mixture models (using Expectation-Maximization (EM)) for clustering and segmentation. Arguably it is association rules that are the only main exception, i.e., association rules are a technique that have no clear "ancestor" in the statistical literature. It is also debatable as to how many successful real-world applications in data mining actually rely on association rules for their success.

Thus, while the annual proceedings of conferences such as SIGKDD and SIGMOD contain many novel and interesting techniques that are not within the mainstream of conventional statistics, when one looks at both current data mining software tools (from the likes of IBM, SAS, SGI, and many others) and at industry-specific applications (such as domain-specific applications developed in e-commerce) there is a heavy reliance on the application of traditional statistical ideas. Indeed, a statistician might argue that data mining is not much more than the scaling up of conventional statistical methods to massive data sets, in effect a large-scale "data engineering" effort.

While there is some truth to this viewpoint, a more accurate reflection of the state of affairs is that data mining (and more generally, computer science) has indeed introduced a number of new ideas within the general realm of data analysis, ideas that are quite novel and distinct from any prior work in statistics. We can identify several such contributions that have arisen primarily from work within computer science (data mining, machine learning, neural networks) rather than from conventional statistics:

1 **Flexible predictive modeling methods**: from the early work on decision trees and neural networks to more recent techniques for combining models, there is a history of computer scientists introducing many new techniques for predictive modeling, particularly for classification problems. These techniques often start out with strong algorithmic foundations but weaker formal statistical justification. Over the past 20 years in machine learning and data mining there has been a recurring pattern of flexible models and algorithms being introduced, developed, and applied by computer scientists (e.g., the boosting framework of Freund and Schapire [FS97]), with the supporting statistical theory being "filled-in" at a later date (e.g., the statistical justification of boosting provided by Friedman, Hastie, and Tibshirani [FHT00]).

2 The use of **hidden variable models** for large-scale clustering and prediction problems. For example, hidden Markov models are an excellent example of how complex non-stationary data (a

speech signal for example) can be "compressed" into a relatively simple hidden state representation that is often quite adequate for classification and prediction applications (see [Ben99] for a review). The EM framework for training hidden variable models such as HMMs is appealing to computer scientists since it is couched as an algorithm, and its algorithmic basis has led to its widespread adoption by computer scientists in a variety of applications (e.g., for model-based collaborative filtering, Heckerman et al., 2000).

3 **Finding patterns** (data mining) rather than global models (statistics): examples of pattern-finding algorithms include association rule algorithms [AIS93], sequential association algorithms [MTI95], rule induction algorithms [WB86, SG92, FF99], and contrast set algorithms [BP99]. These pattern-finding algorithms differ from more conventional statistical modeling in that they do not attempt to "cover" all of the observed data, but instead focus in a data-driven manner on "local" pockets of information.

4 **The engineering of scale**, namely, the data engineering aspects of scaling traditional algorithms to handle massive data sets. Work in this area involves both computationally-driven approaches [ZRL97, ML98, BPR98, GGRL99, PK99] as well as statistically-motivated techniques [DVJ99]. Worth mentioning in this context is the fact that researchers in a variety of areas such as speech, natural language modeling, human genome analysis, and so forth, have all developed a variety of practical learning algorithms and "tricks of the trade" for dealing with massive data sets, e.g., [LB97, WMB99].

5 Analyzing **heterogenous structured data** such as multimedia data (images, audio, video) and Web and text documents. For example, there is a significant research within computer science on using learning algorithms to improve our understanding of the structure of the Web (e.g., [Kle98]) and in learning how topic categories can be automatically extracted from documents (e.g., [Hof99]).

It is important to emphasize that statistics still plays a major role in each of these areas. However, to a large extent it is not statisticians who are leading the charge but rather computer scientists and engineers who are adapting statistical methods to these new problems and opportunities.

There are other general distinctions between data mining and statistics. For example, data mining is typically (indeed, in practice almost always) concerned with observational retrospective data, i.e., data that has already been collected, often for some other purpose (e.g., records of financial transactions recorded by a bank for accounting purposes). Thus, issues such as experimental design (the construction of an experiment to collect data to test a specific hypothesis) are not typically within the vocabulary or tool-set of a data miner. For other general discussions on statistical aspects of data mining see [EP96, GMPS96, GMPS97, HPS97, Han98, Lam00, Smy00].

3. A Reductionist View of Data Mining

Let us consider a very high-level view of data mining and try to reduce a generic data mining algorithm into its component parts. The particular reductionist viewpoint proposed here is not necessarily unique, but it nonetheless does provide some insight into the different and relatively independent components that make up data mining algorithms. As discussed here this breakdown is focused on *algorithms*, rather than the overall data mining *process*: the overall process involves numerous additional (and important) steps such as data selection, data preprocessing, evaluation, and so forth, which we will not discuss in any detail.

In this proposed framework, we identify five primary components of a data mining algorithm (see [HMS01] for further discussion):

1 **The Task:** In data mining we can consider our tasks to fall into a few general categories: exploratory data analysis (e.g., visualization of the data), pattern search (e.g., association rule discovery), descriptive modeling (e.g., clustering or density estimation), and predictive modeling (e.g., classification or regression). This is not perfect but it gives us a high-level categorization of common data mining tasks. An important point is that the task should (if possible) be identified explicitly in any data mining endeavor, since the nature of the task will directly influence the other components described below.

2 **The Model Structure:** This is the specification of the structural form that we seek in the data. For descriptive tasks (e.g., clustering) the structure may be of paramount importance, while in predictive tasks the exact nature of the structure may be only of secondary importance compared to the predictive power of the model. Examples of general structures include decision trees, Gaussian mixtures, association rules, and linear regression models. For example, for a regression problem with 2 input variables X_1 and X_2 and a target variable Y we might propose as a simple linear relationship:

$$Y = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + e \tag{1.1}$$

where e is random additive noise (e.g., Gaussian, zero mean). In this case our model structure is a plane in the two-dimensional X_1, X_2 space with noisy observations (Y values) distributed about this plane. A *specific* model consists of the specification of a particular plane parametrized by $\alpha_0, \alpha_1, \alpha_3$, the three unknown parameters of the structure. Once we specify a particular structural form we effectively limit (for better or worse) our view of the data to a view that is relative to that structure. Naturally, in practice, it is common to specify multiple different types of structures (e.g., both decision tree and logistic regression models for a classification problem) so as to allow the data to indicate which structure is best. We do not need to assume that the true data-generating mechanism is within our model family. Usually our model structures represent convenient approximations to some unknown reality.

3 **The Score Function:** Loosely speaking this is how we evaluate the quality of a particular fitted model or pattern on the basis of observed data, e.g., squared error for regression, classification error for classification, and so forth[1]. The score function provides a link between the hypothesized model structure and the observed data. Naturally, the form of the score function is highly dependent on the nature of the data mining task being undertaken. It is typically defined as a function of the unknown parameters of the model given a particular observed data set. In practice it is customary to choose a score function from a relatively small set of well-known and easily analyzed functional forms, such as squared error for prediction of real-valued quantities, e.g.,

$$S(\alpha_0, \alpha_1, \alpha_2) = \frac{1}{N} \sum_{i=1}^{N} \left(y^i - (\alpha_0 + \alpha_1 x_1^i + \alpha_2 x_2^i) \right)^2 \qquad (1.2)$$

where (y^i, x_1^i, x_2^i) is the ith training tuple, $1 \leq i \leq N$. Thus, we view our score function S as a one-dimensional function of the three unknown α parameters and seek to minimize S given the observed data. In fact we are free to choose a completely general score function to reflect the actual losses (or gains) that would be incurred in practice when using the model, e.g., a general loss matrix for classification rather than simply counting errors. Note that the score function may be used in different ways, e.g., evaluated both on the training data (to search for good parameters) and on a validation or test data set (to search for a good model from a set of fitted potential models, each of which had its parameters fitted on a training data set).

4 **The Optimization Algorithm:** Given a score function, a model structure, and an observed data set, the problem of defining a data mining algorithm can now be reduced to that of defining a computational procedure for finding the extremum of the score function (e.g., minimizing the squared error) over parameter space (and over model space when model complexity is allowed to vary or multiple model forms are present). In a few cases the problem can be solved in closed form. For example, finding the α's that minimize S for the regression problem above can be solved using linear algebra. Similarly, determining the probabilities used in a naive Bayes classifier can be carried out by simply counting relative frequencies of different events. More typical, however, is the case where the score function has a complicated surface as a function of the parameters, with multiple extrema (keep in mind that if we have k parameters then the score function is a scalar function defined over a k-dimensional space). In such cases finding the global extremum of the score function may be an NP-hard problem. Thus, we must resort to some form of heuristic search, e.g., greedy search among the space of all possible decision trees, gradient descent with multiple random restarts in the space of neural network weights, and so forth. This optimization/search component is the heart of the actual data mining algorithm. However, it is important to be aware that a clever optimization algorithm for fitting a model is really only as useful as the model structure, score function, and task that it is based on. In other words, attention needs to be paid that we are in fact asking the right questions of the data (see [Han94] for further discussion in a statistical context).

5 **The Data Management Strategy:** Given an actual algorithm, the final step concerns the implementation of specific mechanisms for how the data are accessed, stored, and managed. In many research papers no explicit data management strategy is specified. For example, most existing machine learning and statistical learning algorithms just assume that the data are read into main memory and reside in a flat file. Data miners are keenly aware that data management can be extremely important in practical applications involving very large data sets. Algorithms such as the A Priori algorithm for association rules make the data management component very explicit in the description of the overall algorithm (i.e., multiple linear scans of the data). Other work takes an existing framework where the model/score function/optimization components have been well-defined in the literature and are tried and

tested in practice, and then embeds them in an explicit data management framework where efficiency of data access is given priority. Examples of this approach include the BOAT framework for efficient decision tree fitting to massive data sets [GGRL99], and the general ADTree data structures of Moore and colleagues for efficient data access and caching for a variety of learning algorithms [ML98, Moo99].

The five-component framework above provides us with a simple systematic language for understanding the "parts" that make up a data mining algorithm. Typically the modeling is guided by concepts from applied mathematics and probability, the score function is often driven by general principles of statistical inference (i.e., how we want to connect an abstract model to actual observed data), and the optimization and data management components are heavily algorithmic in nature. By describing a data mining algorithm within this framework we can understand somewhat more clearly what the primary contribution of each component of a specific algorithm is as well as providing a reference framework for systematically comparing different algorithms and synthesizing new ones.

As an example, data mining algorithms based on either regression trees or neural networks differ primarily in terms of their model structure (hierarchical piecewise constant functions for trees, versus non-linear combinations of weighted sums for networks), as well as in terms of how the optimization is carried out (greedy "one variable at a time" structure search for trees, versus gradient descent in continuous weight-space for networks).

Similarly a general technique such as genetic algorithms is best viewed as belonging to the optimization component of a data mining algorithm, rather than being viewed as an algorithm or model representation unto itself. The nature of the genetic optimization framework imposes representational restrictions on the model structure being used (the model representation must typically be defined as bit-strings in some form). But the primary role of a genetic algorithm is to provide a non-local search strategy in model-space, in contrast to the more traditional local (e.g., gradient-based) search methods. By decoupling the model structure component from the optimization/search component we can mix and match different model structures and optimization techniques in any manner we wish.

Within this framework, we can see that data miners typically place much more emphasis than statisticians on the two primary computational components (optimization and data management). Conversely, statisticians typically place much more emphasis on the model representation and score function components (modeling and inference) than on the computational aspects of how the model is fit. To see examples of this one need only look at typical research papers from each field (e.g., from the *Journal of Data Mining and Knowledge Discovery* and the *Journal of the American Statistical Association*). While this represents two extreme ends of a spectrum (computational issues at one end, modeling/inference at the other) it is nonetheless a useful spectrum along which to consider how data mining and statistics differ. Each community is comfortable with what they know best: computer scientists grow up on a basic diet of algorithms and data structures, while statisticians grow up on a basic diet of mathematical models and principles of statistical inference. Both aspects can play an important role in data mining.

4. An Example: Modeling Transaction Data

Let us briefly consider the problem of analyzing transaction (market-basket) data within the general reductionist framework outlined in the previous section. Association rules are a widely-used technique in data mining for analyzing such data, and are primarily intended for the task of pattern discovery (although one can also consider secondary uses of association rules, such as clustering and prediction). The underlying association rule "structure" is a set of rules, statements of the form "IF A and B and C THEN D with probability 0.9" where A, B, C, and D are all binary propositions (true or false). The emphasis in association rule algorithms is not so much on the nature of the task or the model representation, but more on the search and data management aspects of the algorithm. The essence of the idea is to use fast counting and breadth-first systematic search to find the set of association rules that are above certain pre-defined constrains on support, $p(A, B, C, D)$, and confidence, $p(D|A, B, C)$. Papers on association rule algorithms tend to emphasize computational efficiency of the algorithm itself (the efficiency of the search and data management component) rather than interpretation or evaluation of the rule sets that are obtained.

Association rule algorithms are a good reflection of the database-oriented view of data mining, where the problem of finding all association rules satisfying given confidence and support thresholds can be viewed as a general form of a database query. This contrasts sharply with the statistical view, which places more emphasis on the question of "what to compute," whereas the database researcher places emphasis on the question of "how to compute." For very large data sets in particular, both viewpoints can be thought of as equally valid and important.

As an example, for transaction data, a statistical modeller might propose a probabilistic model for how purchases are made. The arrivals of an individual shopper to a store could be modeled as coming from a Poisson distribution (a constant arrival rate λ over time), where each shopper might have their own individual Poisson rate λ_i. As an extension one might perhaps allow the λ_i's to vary seasonally and/or in a non-stationary manner. Conditioned on the event of having arrived at the store, a shopper's purchase choice of n specific items from the k possible items could be modeled as n independent multinomial (k-way) trials, where n is sampled from a distribution on how many items are in a basket. The multinomials and market-basket size distributions could be modeled as being different for each individual, but where the distribution of parameters across individuals is constrained in some manner.

For small numbers of items in a specific category (e.g., modeling individual purchasing patterns of a particular brand of coffee) this type of Poisson-multinomial model has in fact been well-studied in the marketing literature (e.g., see [GEC84] for a seminal paper on this topic, and [WK98] for a comprehensive recent review). While such models inevitably are built conditioned on various assumptions (such as an assumed independence of the items being purchased from one store visit to the next), this "generative modeling" approach nonetheless can be quite powerful. It provides a systematic framework for hypothesis testing, clustering and segmentation, customer profiling, and forecasting (again see [WK98] for numerous practical examples, and [CGS00] for a general discussion of such generative models for clustering individuals).

In contrast to the association rule framework, this statistical approach to modeling of transaction data focuses strongly on the construction of a suitable data-generating model, with less emphasis on the computational aspects of how the model is fit to the data. Another difference between the two approaches is the fact that association rules look for *patterns* in the data, whereas the statistical model tries to construct a *global model* for the whole data set. Searching for patterns may be advantageous if that is the primary goal and if constructing a global model is difficult or unreasonable.

The main point in comparing the statistical and association rule approach to transaction data is that statistical models provide a flexible and general language for modeling such data, allowing aspects such as individual-level variations, temporal dependencies, and so forth, to be incorporated in a natural and systematic manner. These statistical techniques can be viewed as providing a useful and complementary set of tools to the more computationally-motivated data mining techniques such as association rules. For example, the statistical approach is based on certain modeling assumptions that may not hold in practice. In contrast, association rules are relatively "non-parametric" in the sense that there are few or no constraints assumed about the functional form of the data-generating process.

5. The Role of Statistical Education in Data Mining

In order to appreciate the fundamental role of statistics in data analysis, a data miner needs at least some minimal exposure to statistical concepts. Rather than learning a set of specific detailed models it is probably more important to appreciate the general mindset of statistical thinking, such as the explicit recognition of uncertainty in a broad data analysis context.

Many computer scientists and engineers engaged in data mining have had only a limited exposure to statistical ideas in their undergraduate curriculum (although engineers often receive more statistical education than computer scientists in most countries). While the value of statistical concepts are widely recognized, modern engineering and computer science curricula tend to be very crowded, leaving little room for anything other than a cursory introduction to statistics. For many graduates their only exposure to statistics consists largely of a cookbook-style class on how to use hypothesis tests (see [Lam00] for further elaboration on this topic).

There are at least three fundamental and important ingredients that are often missing from the statistical education of a typical data miner:

1 **Modeling Languages:** The first ingredient is actually not statistics at all, but rather mathematical and probabilistic modeling, languages for constructing explicit models for how the observed data might be generated. For example in our modeling of transaction data earlier, we proposed a constant arrival rate for shopper arrivals (modeled as a Poisson distribution) and a multinomial choice model for item choice. These are the types of basic building blocks that allow a statistical modeller to construct a relatively complex model from simpler functional forms. In terms of our reductionist viewpoint, this mathematical modeling provides the basis for determining the *model structure* component of an algorithm. The models may of course be informed by prior knowledge about the problem. A wide-ranging and widely used standard introductory text for this form of stochastic modeling is [Ros00]. [ED96] also provide a very readable introduction to a broad range of modeling methods in applied multivariate statistics, a topic of

particular relevance to data mining. These are just two of many excellent texts on modeling: for example, there are many other more specialized texts for specific classes of models, such as time-series models and spatial models.

2 **Inference from Data:** The second primary ingredient provides the foundations of inference: how to reason from the observed data to make inferences about the hypothesized model structures. In our reductionist framework, inference tells us (a) how to construct specific score functions from general statistical and decision-theoretic principles, (b) what the general and desirable properties of these score functions are, and (c) what these score functions can then tell us about generalization to new data. The score functions connect the hypothesized model to the observed data. To understand the principles of inference requires a foundation in the basic principles of mathematical statistics: expectation, parameter estimation, likelihood, sufficient statistics, and so forth. While it can be argued that an excessive emphasis on mathematical formalisms may not be productive in the context of the practicalities of data mining, these concepts and methods are essential if a data miner is to be equipped to penetrate the dense jungles of research publications and texts in statistics. In other words, some minimal knowledge of the basic concepts is required so that a data miner can access the many rich and foundational ideas on data analysis that the statistical community has provided down through the ages (and continues to provide). For readers interested in embarking on this journey, [Kni00] provides and up-to-date and readable introduction to mathematical statistics. Other well-known texts that are widely used in academic courses on mathematical statistics include [BD77, HC78, CB90].

3 **The Practical Art of Statistics:** While the first two educational components provide the raw mathematical tools of statistics, the third component provides the *art*, i.e., the art and craft of knowing how the analytical and modeling tools should be applied in practice. This is a much more subtle issue. Indeed it is fair to say that statisticians themselves are often not exposed to this "art" during their undergraduate education but tend instead to acquire it through years of experience. There are different levels to this "art." One level is the ability to know how to construct analytical models for a problem, e.g., the problem of modeling the time-dependent rate of arrival of customers at an e-commerce site or predicting the rate at which customers buy a given product as

a function of demographics. Typically this is done by having access to a "toolbox" of basic functional forms and models. In the e-commerce case we might model the "probability of buying" by some form of logistic regression, i.e., where

$$\log \frac{p(buy|X_1, \ldots, X_d)}{1 - p(buy|X_1, \ldots, X_d)} = \alpha_o + \sum_{i=1}^{d} \alpha_i X_i, \tag{1.3}$$

where the X_i's are different demographic variables such as age, income, etc., and the αs are again parameters of this logistic model that are estimated from the available data.

It is fair to say that of the three basic educational ingredients above, the most difficult one to learn is (not surprisingly) the "art." For example, the ability to successfully build complex models from basic functional forms is deceptively simple, yet it is often glossed over in the statistical literature, which provide little guidance to the uninitiated as to why certain types of models are preferred over others for a given application (with some notable exceptions such as [CS81, Cha95]). It is important to realize that much of statistical analysis is predicated on the use of an assumed model for how the data are generated. Even in non-parametric analyses (i.e., modeling techniques such as trees which rely on very few functional assumptions) there may nonetheless be various implicit assumptions present, such as assuming that the order in which individual observations are presented is irrelevant from the point of view of the model.

Another significant aspect of the "art" is the ability to dissect a problem to obtain insight into why a particular method works well or how it could be improved. Again, the techniques used are often relatively straightforward, but it is often not so obvious to know which technique to apply in which situation. For example, a well-known "dissection" technique in statistics is that of *diagnostics*, namely, taking a model apart to try to find out where its limitations are. In regression, for example, the prediction errors (the residuals) may be plotted or analyzed as function of a specific input variable X_i. If there is evidence of a systematic dependence (e.g., larger errors for larger values of X_i) then it may suggest how the model can be improved. Statisticians traditionally rely on a variety of such diagnostic techniques (including visualization) to provide insight and feedback in an iterative manner: model, fit, evaluate, model, fit, evaluate, and so forth. In the "massive data world" of data mining it is not always practical to apply such diagnostics, yet there is clearly substantial room for improvement in developing tools and techniques that allow a user to interact with both data and models in an iterative manner. In an ideal world this is certainly preferable to treating each data mining algorithm as a stand-alone black-box where the actions of the user are limited to tweaking various algorithmic parameters such as search options and so forth, rather than being allowed to explore the interaction of the data and model more directly.

6. Success Stories from the Interface

There can be significant differences between a typical computer scientist's view of data and a typical statistician's view of the same data. Despite this (or indeed perhaps because of this difference) there are numerous well-known examples of symbiosis at computer science/statistics interface. For example, there appears to be a common recurring pattern where research on essentially the same idea is carried out independently within statistics and within computer science, and both sets of ideas are then subsequently integrated to form a much richer and broader framework. We discuss briefly below a few well-known recent examples of this pattern in the general context of machine learning and data mining:

- Early work on neural networks focused largely on representational properties and biological plausibility of the models and details of training algorithms [RM86]. A broader statistical view of neural networks as highly flexible non-linear regression models gradually began to emerge, both from within the neural network community (e.g., [Bis95]) and from members of the statistical community who had taken an interest in this new form of regression model [GBD92, CT94, Rip94]. For example, links to more established statistical ideas such as generalized linear models and projection pursuit regression led to new models and algorithms that are "hybrids" of statistical and neural network research (e.g., [JJ94]).

- Graph-based models for efficient representation of multivariate distributions had been known for some time in areas such as genetics (e.g., [CTS78]). In the late 1980's more general and widely-applicable frameworks were developed independently within statistics (e.g., the acyclic directed graphical model framework of [LS88]) and the largely equivalent belief network framework pioneered by Pearl [Pea88] within computer science. The last 10 years has seen substantial research activity within both communities, particularly in the area of learning such models directly from data (e.g., [Hec95, Jor98]). The recent work on learning graphical models provides an excellent example of the symbiosis of computer science and statistics: efficient graph-based algorithms for computational inference (computer science) coupled with general principles for parameter and structure estimation from data (statistics).

- Latent (hidden) variable models have a long history in statistics (e.g., see [DT99]). Work on this topic within computer science is more recent, but has had a broad impact in the last 20 years in a variety of applications: hidden Markov models in speech recognition, latent variable models in language and text modeling, mixture-models and clustering techniques, and neural networks with hidden variables [HS99]). The EM procedure, an inherently computational concept but one that is motivated by some very fundamental principles in statistical inference, has played a pivotal role in all of these computer science and engineering applications.

- Decision trees were originally developed in applied statistics in the 1960's (e.g., [MS63]) but did not receive much attention within statistics until the publication of the pioneering work on CART [BFOS94]. Quinlan independently popularized the use of trees in machine learning with his ID3 and C4.5 family of algorithms [Qui87, Qui93]. Both the CART and ID3/C4.5 approaches share many common ideas, resulting in quite similar tree-learning algorithms. The statistical work on trees typically emphasizes parameter estimation and tree selection aspects of the problem, while more recent work on trees in data mining has emphasized data management issues (e.g., [GGRL99]). Decision trees are now a cornerstone in the toolbox of every data miner, with many relatively minor variants of tree algorithms, but all having much in common with the original ideas from the CHAID, CART, and ID3 research.

- Boosting algorithms are a class of techniques for iteratively constructing more complex predictive models by combining simpler models. Boosting was originally proposed within the framework of computational learning theory [FS97], a sub-field of machine learning concerned with theoretical analyses of learning algorithms. (Although having much in common, computational learning theory differs in many aspects from statistics, for example, emphasizing worst-case distribution-free learning models in contrast to the more typical average-case learning models in statistics). Subsequent empirical work revealed that boosting was not only of theoretical interest, but that it provided a very useful practical tool for improving the generalization performance of a wide variety of prediction models [BK99]. Statisticians began to take an interest in boosting, resulting both in powerful new algorithms [Fri99, Rid00] as well a general statistical theory that explains when and why the method works [FHT00].

The point of the examples above is to show that there is a long and successful tradition of "marrying" ideas, theories, and techniques developed relatively independently within computer science and within statistics. Naturally since computer science is a much younger discipline than statistics, the field of statistics has a much broader scope (in the context of learning from data). For example, there are large areas of data analysis such as spatio-temporal modeling, repeated measures/longitudinal data, and so forth, where data mining and machine learning have not had any appreciable impact. On the other hand, there are areas where a computational approach to learning has added concepts to data analysis that are relatively unrelated to anything in statistics. For example, Vapnik's theory of generalization based on margins [Vap98], and the subsequent development of support vector machines based on this theory [SBS99] could be viewed as being quite distinct from conventional statistical thinking on how to build predictive models. Despite these differences, the two fields nonetheless have much in common and data mining can prosper by cultivating and harvesting ideas at the interface.

7. The Dark Side of the Interface

A chapter on data mining and statistics would not be complete without reference to the original use of the term "data mining" within statistics. For a statistician (and particularly one in an area such as econometrics) the term data mining may conjure up a very different perspective than the one that computer scientists are familiar with. Historically in statistics the term "data mining" was used to describe the use of computational search techniques to overfit data sets and uncover patterns of spurious origin (e.g., [Arm67, Lov83]). Terms such as "data snooping," "data dredging," and "data fishing" are all used in the same general context, with clearly negative connotations [SS66, Whi01], e.g., to quote [STW99]:

> Data snooping occurs when a given set of data is used more than once for the purposes of inference or model selection. When such data reuse occurs, there is always the possibility that any satisfactory results may simply be due to chance rather than to any merit inherent in the method yielding the results.

7.1. Variable Selection for Regression

A classic example of data mining in this context occurs in variable selection for regression (e.g., [Mil90]), especially when applied to a relatively small training data set with no data used for holdout testing. Lets say we are trying to fit a simple linear regression model of the form

$$Y = \alpha_0 + \alpha_1 X_i + \alpha_2 X_j + e, \quad 1 \le i, j \le d, \quad i \ne j, \tag{1.4}$$

where Y is the real-valued variable we are trying to predict, the α's are (unknown) parameters of the model, e is additive noise, and X_i and X_j are two particular variables from a set of d variables available to us. So the problem is to find the "best" pair of variables from the set of all possible pairs in the set of d variables. Here "best" is judged by minimum squared error (or some such similar score function) between the model's predictions and the observed Y's. Let ρ be the empirical linear correlation coefficient between the model's predictions and the actual observed Y values, i.e.,

$$\rho = \frac{\frac{1}{n}\sum_{l=1}^{n}(y_l - \hat{y}_l)^2}{\sigma_y \sigma_{\hat{y}}}$$

where y_l is the target value for the lth training data point and $\hat{y}_l$ is the model's prediction based on the lth set of input values, $1 \le l \le n$, and where σ_y and $\sigma_{\hat{y}}$ are the standard deviations of the target values and the predictions respectively. ρ close to 1 (or -1) indicates high positive (or negative) linear correlation, and ρ close to 0 indicates little or no correlation. We can rephrase our problem as that of finding X_i and X_j with the highest positive correlation, i.e., $(i, j) = \arg\max_{i,j}\{\rho_{ij}\}$, where ρ_{ij} is the correlation between y and $\hat{y}$ using the best-fitting linear predictor with variables X_i and X_j. From a computational viewpoint searching for the best pair is an interesting search problem and systematic search strategies such as branch-and-bound can be quite useful.

However, ignoring the details of how the search would be performed, consider the effect of the size of d, the total number of variables available. As d increases, the chances of selecting a pair of variables that *appear* to predict Y well, but in fact are not good predictors, also increases. To see why this can happen, consider for example the case where none of the possible pairs of X variables have any predictive power at all! i.e., all predictions are essentially random noise or equivalently $p(Y|X_i, X_j) = p(Y), \forall i, j$. Of course as data analysts we do not actually know (or even believe) that this is the case, since if we really believed there was no dependence we would not be trying to fit a predictive model. Nonetheless most data miners who have worked on real-world data sets are well aware that we occasionally encounter applications where the available predictor variables appear to have virtually no predictive power.

In the context of such a data set, a significant problem lies in the fact that ρ_{ij} will vary across all (i, j) pairs and will vary from data sample to data sample. Thus, the *maximum* of the empirically observed ρ_{ij}'s is also a random quantity with its own distribution, given a particular data set. Even if the true expected value across different data sets of each individual ρ_{ij} is zero (i.e., the model has no predictive power for each i and j), the expected value of the empirically observed *maximum* over the set of ρ_{ij}'s will be non-zero and positive (i.e., it will be optimistically biased). As the size of this set increases (as the number of candidate variables d is increased) the expected value of the maximum will also increase, i.e., we will tend to find better and better-looking regressions even though no real predictive power exists. This specific problem of spurious correlations has received particular attention in the fields of social science and econometrics [Ein72, Lea78, Hen95].

Other related (but more subtle) versions of "data-dredging" also exist. Imagine, for example, that we have multiple different data sets available to us, perhaps different derived sets of features, sets of features derived for different time-windows (for time-dependent variables), and so forth. Once again, imagine that none of these features have any predictive power. By sequentially fitting models and performing model selection on these different sets of features, we gradually increase our chances of accepting a model that happens to look like it has good predictive power, when in fact it has none.

7.2. Model Selection using Validation Data Sets

Now of course most current data miners are well aware of these sorts of problems. For example, the use of validation data sets and cross-validation techniques are widely used in data mining to provide more accurate estimates of a model's true generalization capabilities. Nonetheless, similar problems still lurk in the background, even with the use of independent training and validation data sets. For example, consider a simple binary classification problem, with N examples, d binary input variables (attributes) $X_1, \ldots, X_d$, and a single binary (target) class variable C. Say we train a variety of different well-known classification models $M_1, \ldots, M_m$ (e.g., decision trees, support vector machines, naive Bayes) with all kinds of variations (boosted versions, different feature subsets, combined models, etc)—modern data mining and statistical packages make it relatively easy to train many different models.

Naturally the models with more degrees of freedom will tend to fit the training data better, so classification error on the training data is not a good guide to how well the model will perform on new data. Since in many data mining applications we are fortunate to have large data sets it is easy to keep aside a separate validation data set for model selection. We estimate the classification error of each of our m models on the validation data, $e_1, \ldots, e_m$, and select the model with the lowest error rate $e_{\min} = \min\{e_1, \ldots, e_m\}$ as the model we will use for future predictions.

Now let us assume that there is no predictive power at all in any of the d input variables, so that one can do no better than random predictions with *any* classifier using these same variables as inputs, i.e., the error rate $= 0.5$ (assuming the two classes are equally likely). This "best achievable" error rate is known as the Bayes error rate for a given set of input variables. Note again that $e_{\min}$ is a random quantity in the sense that from one data set to the next it will vary. $e_{\min}$ for a particular pair of training/validation data sets and a particular set of m classifiers will typically be less than 0.5, and perhaps significantly less than 0.5 depending on how many models we try and how many data points are available. We can think of each of the m classifiers as simply adding in a column of predicted Y values for the validation data set. These columns are not independent since they all are based on models built on the same training data, but to a first approximation can be thought of as being independent if the true underlying relationship between the inputs X and the class variable C is random. We can imagine sorting the observed class values in the validation data set into two subsets of rows, subset A where C takes value 0 and subset B where C takes value 1. Keeping in mind that each of our trained classifiers will be no better than black boxes that randomly produce 0's and 1's (with about equal probability), the validation error rate for each model is simply the number of 1's produced in subset A and the number of 0's produced in subset B, divided by the total number of rows N_v. Essentially we can view this as a version of a binomial random process, no different that tossing a coin N_v times and counting the occurrence of 0's and 1's in certain locations. As N_v goes to infinity, we know by the law of large numbers that indeed the empirically-observed error rate for each column will get closer to the true value of 0.5. But keep in mind that (a) we only have a finite amount of data N_v for validation, and (b) that we are in fact selecting the column with the *minimum* empirical error rate. Clearly, this minimum error rate will tend to be less than 0.5 (i.e., it is optimistically biased) and the expected value of this negative bias (if we repeated this experiment many times) will be a function of both N_v and m, the number of models being used. In extreme cases (small N_v and large m) it could be very biased, e.g., we might find a model with validation error of only 0.3 when the true error rate of all models is 0.5.

Now of course in practice the inputs will usually have *some* predictive power. However, the validation estimate will still be optimistically biased, precisely because it is defined as the minimum over a set of random quantities. The point here is that while validation data and holdout data provide useful "insurance policies" that usually provide much better guidance than performance on training data alone, the element of chance and variability is nonetheless still always present. A useful exercise is to always imagine what your data mining algorithm might do if it were presented with truly random data (as in the examples above). For example, imagine what might happen if a standard association rule algorithm were run on data where all the items (columns) were generated in an independent manner, i.e., there are truly no rules relating any of the items. The exact number of rules found and their strengths will depend on how many items p and how many transactions n there are, but the general effect is the same: even on purely random data, the effect of massive search will be to find spurious associations. Note that although we might use hypothesis testing techniques to guard against such "noise," repeated application of multiple hypothesis tests on a massive scale will eventually lead to acceptance of false hypotheses. It is the massive search (the massive systematic search through itemsets) that is the root cause of this fundamental inability to distinguish pure noise from real structure in the data.

In practice we know that real structure often does exist (i.e., we are not dealing with random data) and that this will tend to dominate the random noise. Nonetheless the general point is that the use of unfettered search techniques can in theory get a data miner into trouble. It is the wise data miner that is aware of these risks and conveys a similar note of caution to the *user* of his/her algorithms, such as the unsuspecting business person or scientist who uses a data mining algorithm in "black box" fashion. Data miners need to be aware of the fundamental statistical aspects of inference from data (see [Jen91, Sal97, JC00] for further discussion). Data mining algorithms should not be a substitute for statistical common sense.

8. Conclusions

This chapter has argued that statistical and algorithmic issues are both important in the context of data mining. We have seen that there is a substantial history of successful research at the interface of computer science and statistics, despite the fact that computer scientists and statisticians have substantially different "cultural biases" in terms of how they think about data. For data miners the message is clear: statistics is an essential and valuable component for any data mining exercise. The future success of data mining will depend critically on our ability to integrate techniques for modeling and inference from statistics into the mainstream of data mining practice.

Acknowledgements

The author gratefully acknowledges valuable feedback on earlier drafts of this chapter from Tom Dietterich and Greg Ridgeway as well as from two anonymous reviewers. This work was supported in part by the National Science Foundation under Grant IRI-9703120.

Notes

1. Note that "score" is used in statistics in a much more specific sense, namely, the derivative of the log-likelihood: here we purposely use "score" in a much broader sense. Readers familiar with this usage might wish to replace "score function" with "loss function" throughout

References

[AIS93] Agrawal, R., Imielinski, T., and Swami, A. (1993) Mining associations between sets of items in massive databases, in *Proceedings of the 1993 ACM SIGMOD International Conference on the Management of Data*, New York, NY: ACM Press, 207–216.

[Arm67] Armstrong, J. S., (1967) Derivation of theory by means of factor analysis or Tom Swift and his electric factor analysis machine, *American Statistician*, 21, 415–22.

[BK99] Bauer, E. and Kohavi, R. (1999) An empirical comparison of voting classification algorithms: bagging, boosting, and variants, *Machine Learning*, 36(1/2), 105–139.

[BP99] Bay, S. and Pazzani, M. (1999) Detecting change in categorical data: mining contrast sets, in *Proceedings of the Fifth ACM International Conference on Knowledge Discovery and Data Mining*, New York, NY: ACM Press, 302–305.

[Ben99] Bengio, Y. (1999) Markovian models for sequential data, *Neural Computing Surveys*, 2, 129–162.

[BL00] Berry, M. J. A. and Linoff, G. (2000) *Mastering Data Mining: The Art and Science of Customer Relationship Management*, New York, NY: John Wiley and Sons.

[BD77] Bickel, P. J. and Doksum, K. A. (1977) *Mathematical Statistics: Basic Ideas And Selected Topics*, San Francisco, Holden-Day.

[Bis95] Bishop, C. (1995) *Neural Networks for Pattern Recognition*, Oxford, UK: Clarendon Press.

[BPR98] Bradley, P., Fayyad, U. M., and Reina, C. (1998) Scaling EM (expectation-maximization) to large databases, Technical Report MSR-TR-98-35, Microsoft Research, Redmond, WA.

[BFOS94] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J., (1984) *Classification and Regression Trees*, Belmont, CA: Wadsworth Statistical Press.

[CGS00] Cadez, I., Gaffney, S., and Smyth, P. (2000) A general probabilistic framework for clustering individuals, in *Proceedings of the ACM Seventh International Conference on Knowledge Discovery and Data Mining*, New York, NY: ACM Press, 140–149.

[CTS78] Cannings, C., Thompson, E. A., and Skolnick, M. H. (1978) Probability functions on complex pedigrees, *Advances in Applied Probability*, 10, 26–61.

[CB90] Casella, G. and Berger, R. L. (1990) *Statistical Inference*, Wadsworth and Brooks.

[Cha95] Chatfield, C. (1995) *Problem Solving*, 2nd ed., Chapman and Hall.

[CT94] Cheng, B. and Titterington, D. M. (1994) Neural networks: a review from a statistical perspective, *Statistical Science*, 9, 2–54.

[CS81] Cox, D. R. and Snell, E. J. (1981) *Applied Statistics: Principles and Examples*, London: Chapman and Hall.

[DH00] Domingos, P. and G. Hulten (2000) Mining high-speed data streams, in *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*, pp. 71-80, Boston, MA: ACM Press.

[DVJ99] Du Mouchel, W., Volinsky, C., Johnson, T., Cortes, C., and Pregibon, D. (1999) Squashing flat files flatter, in *Proceedings of the Fifth ACM International Conference on Knowledge Discovery and Data Mining*, New York, NY: ACM Press, 6–15.

[DT99] Dunmur, A. P. and Titterington, D. M. (1999) Analysis of latent structure models with multidimensional latent variables, in *Statistics and Neural Networks : Advances at the Interface*, J.W. Kay and D.M. Titterington (eds.), New York: Oxford University Press, 165–194.

[Ein72] Einhorn, H. (1972) Alchemy in the behavioral sciences, *Public Opinion Quarterly*, 36, 367–378.

[EP96] Elder, J. F., and Pregibon, D. (1996) A statistical perspective on knowledge discovery in databases, in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, eds., Cambridge, MA: The MIT Press, pp. 83–115.

[ED96] Everitt B. and Dunn G. (1996) *Applied Multivariate Data Analysis*, New York, NY: John Wiley and Sons.

[FS97] Freund, Y. and Schapire, R. (1997) A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, 55(1), 119–139.

[Fri99] Friedman, J. H. (1999) Greedy function approximation: a gradient boosting machine, Technical Report, Statistics Department, Stanford University.

[FF99] Friedman, J. H. and Fisher, N. I. (1999) Bump hunting in high-dimensional data, *Statistics and Computing*, 9, 123–143.

[FHT00] Friedman, J. H., Hastie, T., and Tibshirani, R. (2000), Additive logistic regression: a statistical view of boosting, *Annals of Statistics*, to appear.

[GGRL99] Gehrke, J., Ganti, V., Ramakrishnan, R., Loh, W-Y. (1999) BOAT—Optimistic decision tree construction. *Proceedings of the SIGMOD Conference 1999*, New York, NY: ACM Press, 169–180.

[GBD92] Geman, S., Bienenstock, E., and Doursat, R. (1992) Neural networks and the bias/variance dilemma, *Neural Computation*, 4, 1–58.

[GMPS96] Glymour C., Madigan D., Pregibon D., Smyth P. (1996) Statistical inference and data mining, *Communications of the ACM*, 39(11), 35–41.

[GMPS97] Glymour C., Madigan D., Pregibon D., Smyth P. (1997) Statistical themes and lessons for data mining, *Journal of Data Mining and Knowledge Discovery*, 1, 11–28.

[GEC84] Goodhardt, G. J., Ehrenberg, A. S. C., and Chatfield, C. (1984) The Dirichlet: a comprehensive model of buying behavior, *J. R. Statist. Soc. A*, 147(5), 621–655.

[Han94] Hand, D. J. (1994) Deconstructing statistical questions, *J. R. Statist. Soc. A*, 157(3), 317–356.

[Han98] Hand D. J. (1998) Data mining—statistics and more, *The American Statistician*, 52(2), 112–118.

[HMS01] Hand, D. J., Mannila, H., and Smyth, P. (2001) *Principles of Data Mining*, Cambridge, MA: The MIT Press, forthcoming.

[Hec95] Heckerman, D. (1995) A tutorial on learning Bayesian networks, Techical Report MSR-TR-95-06, Microsoft Research, Redmond, WA.

[HCM00] Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., and Kadie, C. (2000) Dependency networks for density estimation, collaborative filtering, and data visualization, Technical Report MSR-TR-2000-16, Microsoft Research, Redmond, WA.

[Hen95] Hendry, D. F. (1995) *Dynamic Econometrics*, New York, NY: Oxford University Press.

[HS99] Hinton, G.E., and T. Sejnowski (eds.) (1999) *Unsupervised Learning: Foundations of Neural Computation*, The MIT Press.

[Hof99] Hoffmann, T. (1999) Probabilistic latent sematic indexing, *Proceedings of SIGIR '99*, 50–57.

[HC78] Hogg, R. V. and Craig, A. T. (1978) *Introduction to Mathematical Statistics*, 4th ed. Macmillan.

[HPS97] Hosking, J. R. M., Pednault, E. P. D., and Sudan, M. (1997) A statistical perspective on data mining, *Future Generation Computer Systems*, 13, 117-134.

[Jen91] Jensen, D. (1991) Knowledge discovery through induction with randomization testing, in *Proceedings of the 1991 Knowledge Discovery in Databases Workshop*, G. Piatetsky-Shapiro (ed.), Menlo Park, CA: AAAI Press, 148–159.

[JC00] Jensen, D. and Cohen, P. (2000) Multiple comparisons in induction algorithms, *Machine Learning*, 38, 309–338.

[Joh99] John, G. (1999) personal communication.

[JJ94] Jordan, M. I. and Jacobs, R. A. (1994) Hierarchical mixtures of experts and the EM algorithm, *Neural Computation*, 6, 181–214.

[Jor98] Jordan, M. I. (ed.) (1998) *Learning in Graphical Models*, Cambridge, MA: The MIT Press.

[Kle98] Kleinberg, J. M. (1998) Authoritative sources in a hyperlinked environment, in *Proc. of ACM-SIAM Symp. on Discrete Algorithms*, 668–677.

[Kni00] Knight, K. (2000) *Mathematical Statistics*, Chapman and Hall.

[Koh00] Kohavi, R. (2000) personal communication.

[Lam00] Lambert, D. (2000) What use is statistics for massive data?, preprint.

[LS88] Lauritzen, S. L. and Spiegelhalter, D. J. (1988) Local computations with probabilisties on graphical structures and their application to expert systems (with discussion), *J. Roy. Statist. Soc. B*, 50, 157–224.

[Lea78] Leamer, E. E. (1978) *Specification Searches: Ad Hoc Inference with Non-Experimental Data*, New York, NY: John Wiley.

[LB97] Letsche, T. A. and Berry, M. W. (1997) Large-scale information retrieval with latent semantic indexing, *Information Sciences—Applications*, 100, 105–137.

[Lov83] Lovell, M. (1983) Data mining, *Review of Economics and Statistics*, 65, 1–12.

[MTI95] Mannila, H., Toivonen, H. and Inkeri Verkamo, A. (1995) Discovering frequent episodes in sequences, in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, Menlo Park, CA: AAAI Press, 210–215.

[Mil90] Miller, A. J. (1990) *Subset Selection in Regression*, London, Chapman and Hall.

[Moo99] Moore, A. W., (1999) Cached sufficient statistics for automated discovery and data mining from massive data sources, online white paper, Department of Computer Science, Carnegie Mellon University, July 1999.

[ML98] Moore, A. W. and Lee, M. (1998) Cached sufficient statistics for efficient machine learning with large data sets, *Journal of Artificial Intelligence Research*, 8, 67–91.

[MS63] Morgan, J. N. and Sonquist, J. A. (1963) Problems in the analysis of survey data and a proposal, *J. Am. Stat. Assoc*, 58, 415–434.

[Pea88] Pearl, J. (1988) Probabilistic Inference in Intelligent Systems: Networks of Plausible Inference, San Mateo, CA: Morgan Kaufmann.

[PK99] Provost, F. and Kolluri, V. (1999) A survey of methods for scaling up inductive algorithms, *Journal of Data Mining and Knowledge Discovery*, 3(2), 131–169.

[Qui87] Quinlan, J. R. (1987) Generating production rules from decision trees, *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann, 304–307.

[Qui93] Quinlan, J. R. (1993) *C4.5: Programs for Machine Learning*, San Mateo: CA, Morgan Kaufmann.

[Rid00] Ridgeway, G., (2000) Prediction in the era of massive data sets, *Statistical Modeling for Data Mining*, P. Giudici (ed.), Kluwer, 109–119.

[Rip94] Ripley, B. D. (1994) Neural networks and related methods for classification (with discussion), *J. R. Statist. Soc. B*, 56, 409–456.

[Rip96] Ripley, B. D. (1996) *Pattern Recognition and Neural Networks*, Cambridge, UK: Cambridge University Press.

[Ros00] Ross, S. M. (2000) *Introduction to Probability Models*, San Diego, CA: Academic Press.

[RM86] Rumelhart, D. E. and McClelland, J. L. (eds.) (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*, Cambridge, MA: The MIT Press.

[Sal97] Salzberg, S. L. (1997) On comparing classifiers: pitfalls to avoid and a recommended approach, *Data Mining and Knowledge Discovery*, 1:3, 317–327.

[SBS99] Scholkopf, C., Burges, J. C., and Smola, A. J. (1999) *Advances in Kernel Methods*, Cambridge, MA: MIT Press.

[SS66] Selvin, H. and Stuart, A. (1966) Data dredging procedures in survey analysis, *American Statistician*, 20(3), 20–23.

[SG92] Smyth, P. and Goodman, R. (1992) An information-theoretic approach to rule induction from databases, *IEEE Transactions on Knowledge and Data Engineering*, 4(4), 301–306.

[Smy00] Smyth, P. (2000) Data mining: data analysis on a grand scale?, *Statistical Methods in Medical Research*, 9, 309–327.

[STW99] Sullivan, R., Timmermann, A. and White, H. (1999) Data snooping, technical trading rule performance, and the bootstrap, *Journal of Finance*, 54, 1647–1692.

[Vap98] Vapnik, V. (1998) *Statistical Learning Theory*, New York, NY: Springer Verlag.

[WB86] Walker, M. G. and Blum, R. L. (1986) Towards automated discovery from clinical databases: the RADIX project, in *Proceedings of the Fifth Conference on Medical Informatics*, volume 5, 32–36.

[WMB99] Witten I. H., Moffat A., Bell T. C. (1999) *Managing gigabytes: compressing and indexing documents and images.* San Francisco, CA: Morgan Kaufmann, (2nd ed.).

[WK98] Wedel, M. and Kamakura, W. A. (1998) *Market Segmentation: Conceptual and Methodological Foundations*, Boston, MA: Kluwer Academic Publishers.

[Whi01] White, H. (2001), A reality check for data snooping, *Econometrica*, forthcoming.

[ZRL97] Zhang, T., Ramakrishnan, R., Livny, M., (1997) BIRCH: A new data clustering algorithm and its applications, *Journal of Data Mining and Knowledge Discovery*, 1(2), 141–182.