RC 20856 (92315) 05/21/97
Mathematics 25 pages

Research Report

A Statistical Perspective on Data Mining

Jonathan R. M. Hosking, Edwin P. D. Pednault and Madhu Sudan

IBM Research Division
T. J. Watson Research Center
Yorktown Heights, NY 10598

IBM **Research Division**
Almaden • T.J. Watson • Tokyo • Zurich

A statistical perspective on data mining

Jonathan R. M. Hosking, Edwin P. D. Pednault and Madhu Sudan

IBM Research Division
P. O. Box 218
Yorktown Heights, NY 10598

Abstract. Data mining can be regarded as a collection of methods for drawing inferences from data. The aims of data mining, and some of its methods, overlap with those of classical statistics. However, there are some philosophical and methodological differences. We examine these differences, and we describe three approaches to machine learning that have developed largely independently: classical statistics, Vapnik's statistical learning theory, and computational learning theory. Comparing these approaches, we conclude that statisticians and data miners can profit by studying each other's methods and using a judiciously chosen combination of them.

1. Introduction: a statistician looks at data mining

The recent upsurge of interest in the field variously known as data mining, knowledge discovery or machine learning[1] has taken many statisticians by surprise. Data mining attacks such problems as obtaining efficient summaries of large amounts of data, identifying interesting structures and relationships within a data set, and using a set of previously observed data to construct predictors of future observations. Statisticians have well established techniques for attacking all of these problems. Exploratory data analysis, a field particularly associated with J. W. Tukey [19], is a collection of methods for summarizing and identifying patterns in data. Many statistical models exist for explaining relationships in a data set or for making predictions: cluster analysis, discriminant analysis and nonparametric regression can be used in many data mining problems. It is therefore tempting for a statistician to regard data mining as no more than a branch of statistics.

Nonetheless, the problems and methods of data mining have some distinct features of their own. Data sets can be very much larger than is usual in statistics, running to hundreds of gigabytes or terabytes. Data analyses are on a correspondingly larger scale, often requiring days of computer time to fit a single model. There are differences of emphasis in the approach to modeling: compared with statistics, data mining pays less attention to the large-sample asymptotic properties of its inferences and more to the general philosophy of "learning", including consideration of the complexity of models and of the computations that they require. Some modeling techniques, such as rule-based methods, are difficult to fit into the classical statistical framework, and others, such as neural networks, have an extensive methodology and terminology that has developed largely independently of input from statisticians.

This paper is a brief introduction to some of the similarities and differences between statistics and data mining. In Section 2 we observe some of the differences between the statistical and data-mining approaches to data analysis and modeling. In Sections 3–5 we describe in more detail some approaches to machine learning that have arisen in three more-or-less disjoint academic communities: classical statistics, the statistical learning theory of V. Vapnik, and computational learning theory. Section 6 contains some comparisons and conclusions.

2. Statistics and data mining

2.1. Features of data mining

Both statistics and data mining are concerned with drawing inferences from data. The aim of the inference may be understanding the patterns of correlation and causal links among the data values ("explanation"), or making predictions of future data values

[1]Unfortunately, "data mining" is a pejorative term to statisticians, who use it to describe the fitting of a statistical model that is unjustifiably elaborate for a given data set (e.g. [11]). "Machine learning" is probably better, though "learning" is a loaded term.

("generalization"). Classical statistics has developed an approach, described further in Section 3 below, that involves specifying a model for the probability distribution of the data and making inferences in the form of probability statements. Data-mining methods have in many cases been developed for problems that do not fit easily into the framework of classical statistics and have evolved in isolation from statistics. Even when applied to familiar statistical problems such as classification and regression, they retain some distinct features. We now mention some features of the data-mining approaches and their typical implementations.

Complex models. Some problems involve complex interactions between feature variables, with no simple relationships being apparent in the data. Character recognition is a good example; given a 16×16 array of pixels, it is difficult to formulate a comprehensible statistical model that can identify the character that corresponds to a given pattern of dots. Data-mining techniques such as neural networks and rule-based classifiers have the capacity to model complex relationships and should have better prospects of success in complex problems.

Large problems. By the standards of classical statistics, data mining often deals with very large data sets (10^4 to 10^7 examples). This is in some cases a consequence of the use of complex models, for which large amounts of data are needed to derive secure inferences. In consequence, issues of computational complexity and scalability of algorithms are often of great importance in data mining.

Many discrete variables. Data sets that contain a mixture of continuous and discrete-valued variables are common in practice. Most multivariate analysis methods in statistics are designed for continuous variables. Many data mining methods are more tolerant of discrete-valued variables. Indeed, some rule-based approaches use only discrete variables and require continuous variables to be discretized.

Wide use of cross-validation. Data-mining methods often seek to minimize a loss function expressed in terms of prediction error: for example, in classification problems the loss function might be the misclassification rate on a set of examples not used in the model-fitting procedure. Prediction error is often estimated by cross-validation, a technique known to statistics but used much more widely in data mining.

Minimization of the prediction error estimated by cross-validation is a powerful technique that can be used in a nested fashion—the "wrapper method" [7]—to optimize several aspects of the model. These include various parameters that might otherwise be chosen arbitrarily (e.g., the amount of pruning of a decision tree, or the number of neighbors to use in a nearest-neighbor classifier) and the choice of which feature variables are relevant for classification and which can be eliminated from the model.

Few comparisons with simple statistical models. When data mining methods are used on problems to which classical statistical methods are also applicable, direct comparison of the approaches is possible but seems rarely to be performed. Some comparisons have found that the greater complexity of data mining methods is not always justifiable: Ripley [16] cites several examples. Statistical methods are particularly likely to be preferable when fairly simple models are adequate and the important variables can be identified before modeling. This is a common situation in biomedical research, for example. In this context Vach et al. [20] compared neural networks and logistic regression and concluded that the use of neural networks "does not necessarily imply any progress: they fail in translating their increased flexibility into an improved estimation of the regression function due to insufficient sample sizes, they do not give direct insight to the influence of single covariates, and they are lacking uniqueness and reproducibility".

2.2. Classification: an illustrative problem

A common problem in statistics and data mining is to use observations on a set of "feature variables" to predict the value of a "class variable". This problem corresponds to statistical models for classification when the class variable takes a discrete set of values and for regression when the values of the class variable cover a continuous range. To illustrate the range of approaches available in statistics and data mining we consider the classification problem. Many different methods are used for classification. The classical statistical approach is discriminant analysis; starting from this one can list various data-mining methods in decreasing order of their resemblance to classical statistical modeling. More details of many of these methods can be found in [13]. We denote the class variable by y and the feature variables by the vector $\mathbf{x} = [x_1 \ \dots \ x_f]$. It is sometimes convenient to think of the feature variables as ordinates of a "feature space" with the aim of the analysis being to partition the feature space into regions corresponding to the different classes (values of y).

Linear/quadratic/logistic discriminant analysis. Discriminant analysis is a classical statistical technique based on statistical models containing, usually, relatively few parameters. The modeling procedure seeks linear or quadratic combinations of the feature variables that identify the boundaries between classes. The most detailed theory applies to cases in which the features are continuous-valued and, within each class, approximately Normally distributed.

Projection pursuit. For classification problems, projection pursuit can be thought of as a generalization of logistic discrimination that also involves linear combinations of features but also includes nonlinear transformations of these linear combinations, with the probability of a feature vector $\mathbf{x}$ belonging to class k being modeled as

$$\sum_{m=1}^{M} \beta_{km} \, \psi_m \left(\sum_{j=1}^{f} \theta_{mj} x_j \right) . \tag{1}$$

The ψ_m are prespecified scatterplot smoothing functions, chosen in part for their speed of computation. The nonlinearities and often large numbers of parameters in the model leads one to regard projection pursuit as a "neostatistical" rather than a classical statistical technique.

Radial basis functions. Radial basis functions form another kind of nonlinear neostatistical model. The probability of a feature vector $\mathbf{x}$ belonging to class k is modeled as

$$\sum_{m=1}^{M} \theta_m \phi(||\mathbf{x} - \mathbf{c}_m||/\beta_m)\,.$$

Here $||\mathbf{x} - \mathbf{c}_m||$ is the distance from point $\mathbf{x}$ in feature space to the mth center $\mathbf{c}_m$, β_m is a scale factor, and ϕ is a basis function, often chosen to be the Gaussian function $\phi(r) = \exp(-r^2)$.

Neural networks. A common form of neural network for the classification problem, the multilayer feedforward network, can be thought of as a model similar to (1). However, the ψ_m transformations are different—generally the logistic function $\psi_m(t) = 1/\{1 + \exp(-t)\}$ is used—and more than one layer of logistic transformations may be applied. Neural networks are recognizably close to neostatistical models, but a unique methodology and terminology for neural networks has developed that is unfamiliar to statisticians.

Graphical models. Graphical models, also known as Bayesian networks, belief functions, or causal diagrams, involve the specification of a network of links between feature and class variables. The links specify relations of statistical dependence between particular features; equally importantly, absence of a direct link between two features is an assertion of their conditional independence given the other features appearing in the network. Links in the network can be interpreted as causal relations between features—though this is not always straightforward, as exemplified by the discussion in [15]—which can yield particularly informative inferences. For realistic problems, graphical models involve large numbers of parameters and do not fit well into the framework of classical statistical inference.

Nearest-neighbor methods. At its simplest, the k-nearest neighbor procedure assigns a class to point $\mathbf{x}$ in feature space according to the majority vote of the k nearest data points to $\mathbf{x}$. This is a smoothing procedure, and will be effective when class probabilities vary smoothly over the feature space. Questions arise as to the choice of k and of an appropriate distance measure in feature space. These issues are not easily expressed in terms of classical statistical models. Model specification is therefore determined by maximizing classification accuracy on a set of training data rather than by formally specifying and fitting a statistical model.

Decision trees. A decision tree is a succession of partitions of feature space, each partition usually based on the value taken by a single feature, until the partitions are so fine that each corresponds to a single value of the class variable. This formulation bears little resemblance to classical parametric statistical models. Choice of the best tree representation is obtained by comparing different trees in terms of their predictive accuracy, estimated by cross-validation, and their complexity, often measured by minimum description length.

Rules. Rule-based methods seek to assign class labels to subregions of feature space according to logical criteria such as

$$\text{if } x_1 = 3 \text{ and } x_2 \geq 15 \text{ and } x_2 < 30 \text{ then } y = 1.$$

Individual rules can be complex and hard to interpret subjectively. Rule-generation methods often involve parameters whose optimal values are unknown. The methods cannot be expressed in terms of classical statistical models, and the parameter values are optimized, as for decision trees, by consideration of a rule set's predictive accuracy and complexity.

The foregoing list illustrates a wide range of statistical and data-mining approaches to the classification problem. However, each approach requires at some stage the selection of appropriate features $x_1, \ldots, x_f$. It can be argued that this similarity between the approaches outweighs all of their differences. Any given data set may contain irrelevant or poorly measured features which only add noise to the analysis and should for efficiency's sake be deleted; some dependences between class and features may be most succinctly expressed in terms of a function of several features rather than by a single feature. No method can be expected to perform well if does not use the most informative features: "garbage in, garbage out".

Explicit feature selection criteria have been developed for several of the methods described above. These range from criteria based on significance tests for statistical models to measures based on the impurity of the conditional probability distribution of the class variable given the features, used in decision-tree and rule-based classifiers [10]. As noted above, the "wrapper" method is a powerful and widely applicable technique for feature selection.

Construction of new features can be explicit or implicit. Some techniques such as principal-components regression explicitly form linear combinations of features that are then used as new feature variables in the model. Conversely, the linear combinations $\sum_j \theta_{mj} x_j$ of features that appear in the representation (1) for projection-pursuit and neural-network classifiers are implicit constructed features. Construction of nonlinear combinations of features is generally a matter for subjective judgement.

3. Classical statistical modeling

In this section we give a brief summary of the classical "frequentist" approach to statistical modeling and scientific inference. A detailed account of the theory is given by Cox and Hinkley [2]. The techniques used in applied statistical analyses are described in more specialized texts such as [4] for classification problems and [29] for regression. We assume that inference focuses on a data vector $\mathbf{z}$ with the available data $\mathbf{z}_i, i=1,\ldots,\ell$, being ℓ instances of $\mathbf{z}$. In many problems, such as regression and classification, the data vector $\mathbf{z}$ is decomposed into $\mathbf{z}=[\mathbf{x}, y]$ and y is modeled as a function of the $\mathbf{x}$ values.

3.1. Model specification

A statistical model is the specification of a frequency distribution $p(\mathbf{z})$ for the elements of the data vector $\mathbf{z}$. This enables "what happened" (the observed data vector) to be quantitatively compared with "what might have happened, but didn't" (other potentially observable data vectors).

In regression and classification problems the conditional distribution of y given $\mathbf{x}$, $p(y|\mathbf{x})$, is of interest; the frequency distribution of $\mathbf{x}$ may or may not be relevant. In most statistical regression analyses the model has the form

$$y = f(\mathbf{x}) + e \tag{2}$$

where e is an error term having mean zero and some probability distribution; i.e., it is assumed that the relationship between y and $\mathbf{x}$ is observed with error. The alternative specification in which the functional relationship $y=f(\mathbf{x})$ is exact and uncertainty arises only when predicting y at hitherto unobserved values of $\mathbf{x}$ is much less common: one example is the interpolation of random spatial processes by kriging [8].

In classical statistics, model specification has a large subjective component. Candidates for the distribution of $\mathbf{z}$, or the form of the relationship between y and $\mathbf{x}$, may be obtained from inspection of the data, from familiarity with relations established by previous analysis of similar data sets, or from a scientific theory that entails particular relations between elements of the data vector.

3.2. Estimation

Model specification generally involves an unknown parameter vector $\boldsymbol{\theta}$. This is typically estimated by the maximum-likelihood procedure: the joint probability density function of the data, $p(\mathbf{z};\boldsymbol{\theta})$, is maximized over $\boldsymbol{\theta}$. Maximum-likelihood estimation can be regarded as minimization of the loss function $-\log p(\mathbf{z};\boldsymbol{\theta})$. When the data are assumed to be a set of independent and identically distributed vectors $\mathbf{z}_i$, $i=1,\ldots,\ell$, this loss function is

$$\sum_{i=1}^{\ell} -\log p(\mathbf{z}_i;\boldsymbol{\theta}) .$$

When the data vector is decomposed as $\mathbf{z}=[\mathbf{x}, y]$, the observed data are similarly decomposed as $\mathbf{z}_i=[\mathbf{x}_i, y_i]$, and the loss function (negative log-likelihood) is

$$\sum_{i=1}^{\ell} -\log p(y_i|\mathbf{x}_i;\boldsymbol{\theta}) .$$

If the conditional distribution of y_i given $\mathbf{x}_i$ is Normal with mean a function of $\mathbf{x}_i$, $f(\mathbf{x}_i;\boldsymbol{\theta})$, and variance independent of i, this loss function is equivalent to the sum of squares

$$\sum_{i=1}^{\ell}\{y_i - f(\mathbf{x}_i;\boldsymbol{\theta})\}^2\,.$$

The justification for maximum-likelihood estimation is asymptotic: the estimators are consistent and efficient as the sample size ℓ increases to infinity. Except for certain models whose analysis is particularly simple, classical statistics has little to say about finite-sample properties of estimators and predictors.

Assessment of the accuracy of estimated parameters is an important part of frequentist inference. Estimates of accuracy are typically expressed in terms of confidence regions. In frequentist inference the parameter $\boldsymbol{\theta}$ is regarded as fixed but unknown, and does not have a probability distribution. Instead one considers hypothetical repetitions of the process of generation of data from the model with a fixed value $\boldsymbol{\theta}_0$ of the parameter vector $\boldsymbol{\theta}$, followed by computation of $\hat{\boldsymbol{\theta}}$, the maximum-likelihood estimator of $\boldsymbol{\theta}$. Over these repetitions a probability distribution for $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$ will be built up. Likelihood theory provides an asymptotic large-sample approximation to this distribution. From it one can determine a region $C(\hat{\boldsymbol{\theta}})$, depending on $\hat{\boldsymbol{\theta}}$, of the space of possible values of $\boldsymbol{\theta}$, that contains the true value $\boldsymbol{\theta}_0$ with probability γ (no matter what this true value may be). $C(\hat{\boldsymbol{\theta}})$ is then a confidence region for $\boldsymbol{\theta}$ with confidence level γ. The size of the region is a measure of the accuracy with which the parameter can be estimated.

Confidence regions can also be obtained for subsets of the model parameters and for predictions made from the model. These too are asymptotic large-sample approximations. Confidence statements for parameters and predictions are valid only on the assumption that the model is correct, i.e. that for some value of $\boldsymbol{\theta}$ the specified frequency distribution $p(\mathbf{z};\boldsymbol{\theta})$ for $\mathbf{z}$ accurately represents the relative frequencies of all of the possible values of $\mathbf{z}$. If the model is false, predictions may be inaccurate and estimated parameters may not be meaningful.

3.3. Diagnostic checking

Inadequacy of a statistical model may arise from three sources. Overfitting occurs when the model is unjustifiably elaborate, with the model structure in part representing merely random noise in the data. Underfitting is the converse situation, in which the model is an oversimplification of reality with additional structure being needed to describe the patterns in the data. A model may also be inadequate through having the wrong structure: for example, a regression model may relate y linearly to x when the correct physical relation is linear between $\log y$ and $\log x$.

Comparison of parameters with their estimated accuracy provides a check against overfitting. If the confidence region for a parameter includes the value zero, then a simpler model in which the parameter is dropped will usually be deemed adequate.

In the frequentist framework, underfitting by a statistical model is typically assessed by diagnostic goodness-of-fit tests. A statistic T is computed whose distribution can be found, either exactly or as a large-sample asymptotic approximation, under the assumption that the model is correct. If the computed value of T is in the extreme tail of its distribution there is an indication of model inadequacy: either the model is wrong or something very unusual has occurred. An extreme value of T often (but not always) suggests a particular direction in which the model is inadequate, and a way of modifying the model to correct the inadequacy.

Many diagnostic plots and statistics have been devised for particular statistical models. Though not used in formal goodness-of-fit tests, they can be used as the basis of subjective judgements of model adequacy, for identification either of underfitting or of incorrect model structure. For example, the residuals from a regression model that is correctly specified will be approximately independently distributed; if a plot of residuals against the fitted values shows any noticeable structure, this is an indication of model inadequacy and may suggest some way in which the model should be modified.

Diagnostic plots are also used to identify data values that are unusual in some respect. Unusual observations may be outliers, values that are discordant with the pattern of the other data values, or influential values, which are such that a small change in the data value will have a large effect on the estimated values of the model parameters. Such data points merit close inspection to check whether the outliers may have arisen from faulty data collection or transcription, and whether the influential values have been measured with sufficient accuracy to justify conclusions drawn from the model and its particular estimated parameter values. In analyses in which there is the option of collecting additional data at controlled points, for example when modeling the relation $y = f(\mathbf{x})$ where $\mathbf{x}$ can be fixed and the corresponding value of y observed, the most informative $\mathbf{x}$ values at which to collect more data will be in the neighborhood of outlying and influential data points.

3.4. Model building as an iterative procedure

The sequence of specification–estimation–checking lends itself to an iterative procedure in which model inadequacy revealed by diagnostic checks suggests a modified model specification designed to correct the inadequacy; the modified model is then itself estimated and checked, and the cycle is repeated until a satisfactory model is obtained. This procedure often has a large subjective component, arising from the model specifications and the choice of diagnostic checks. However, formal procedures to identify the best model can be devised if the class of candidate models can be specified *a priori*. This is the case, for example, when the candidates form a sequence of nested models $\mathcal{M}_1, \ldots, \mathcal{M}_m$, whose parameter vectors $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(m)}$ are such that every element of $\boldsymbol{\theta}^{(j)}$ is also included in $\boldsymbol{\theta}^{(j+1)}$. Careful control over the procedure is necessary in order to ensure that inferences are valid, for example that confidence regions for the parameters in the final model have the correct coverage probability.

Classical frequentist statistics has little to say about the choice between nonnested models, for example whether a regression model $y = \theta_1^{(1)} x_1 + \theta_2^{(1)} x_2$ is superior to an alternative model $\log y = \theta_1^{(2)} x_1 + \theta_2^{(2)} x_3$. Such decisions are generally left as a matter of subjective judgement based on the quality of fit of the models, their ease of interpretation and their concordance with known physical mechanisms relating the variables in the model.

Once a satisfactory model has been obtained, further inferences and predictions are typically based on the assumption that the final model is correct. This is problematical in two respects. In many situations one may believe that the true distribution of $\mathbf{z}$ has a very complex structure to which any statistical model is at best an approximation. Furthermore, the statistical properties of parameter estimators in the final model may be affected by the fact that several models have been estimated and tested on the same set of data, and failure to allow for this can lead to inaccurate inferences.

As an example of this last problem, we consider stepwise regression. This is a widely used procedure for identifying the best statistical model, in this case deciding which elements of the $\mathbf{x}$ component of the data vector should appear in the regression model (2). Because random variability can cause x variables that are actually unrelated to y to appear to be statistically significant, the estimated regression coefficients of the variables selected for the final model tend to be overestimates of the absolute magnitude of the true parameter values. This "selection bias" leads to underestimation of the variability of the error term in the regression model, which can lead to poor results when the final model is used for prediction. In practice it is often better to use all of the available variables rather than a stepwise procedure for prediction [14].

3.5. Recent developments

Developments in statistical theory since the 1970s have addressed some of the difficulties with the classical frequentist approach. Akaike's information criterion [18], and related measures of Schwarz and Rissanen, provide likelihood-based comparisons of nonnested models. Such comparisons can be systematically incorporated into an approach to statistical modeling that seeks a minimal-length description of the data: a "description" involves both a model that summarizes the data and the data's deviations from the model. This approach, known as MDL (minimum description length) or MML (minimum message length) has developed a large literature: here we mention only two of the principal references, [17] and [28]. Development of robust estimators [6] has made inference less susceptible to outliers and influential data values. Greater use of nonlinear models enables a wider range of $\mathbf{x}$–y relationships to be accurately modeled. Simulation-based methods such as the bootstrap [3] enable better assessment of accuracy in finite samples.

4. Vapnik's statistical learning theory

One reason that classical statistical modeling has a large subjective component is that most of the mathematical techniques used in the classical approach assume that the form of the correct model is known and that the problem is to estimate its parameters. In data mining, on the other hand, the form of the correct model is usually unknown. In fact, discovering an adequate model, even if its form is not exactly correct, is often the purpose of the analysis. This situation is also faced in classical statistical modeling and has led to the creation of the diagnostic checks discussed earlier. However, even with these diagnostics, the classical approach does not provide firm mathematical guidance when comparing different types of models. The question of model adequacy must still be decided subjectively based on the judgment and experience of the data analyst.

This latter source of subjectivity has motivated Vapnik and Chervonenkis [25, 26, 27] to develop a mathematical basis for comparing models of different forms and for estimating their relative adequacies. This body of work, now known as statistical learning theory, presumes that the form of the correct model is truly unknown and that the goal is to identify the best possible model from a given set of models. The models need not be of the same form and none of them need be correct. In addition, comparisons between models are based on finite-sample statistics, not asymptotic statistics as is usually the case in the classical approach. This shift of emphasis to finite samples enables overfitting to be quantitatively assessed. Thus, the underlying premise of statistical learning theory closely matches the situation actually faced in data mining.

4.1. Model specification

As in classical statistical modeling, models for the data must be specified by the analyst. However, instead of specifying a single (parametric) model whose form is then assumed to be correct, a series of competing models must be specified one of which will be selected based on an examination of the data. In addition, a preference ordering over the models must also be specified. This preference ordering is used to address the issue of overfitting. In practice, models with fewer parameters or degrees of freedom are preferable to those with more, since they are less likely to overfit the data. When applying statistical learning theory, one searches for the most preferable model that best explains the data.

4.2. Estimation

Estimation plays a central role in statistical learning theory just as it does in classical statistical modeling; however, what is being estimated is quite different. In the classical approach, the form of the model is assumed to be known and, hence, emphasis is placed on estimating its parameters. In statistical learning theory, the correct model is assumed to be unknown and emphasis is placed on estimating the relative performance of competing models so that the best model can be selected.

The relative performance of competing models is measured through the use of loss functions. The negative log-likelihood functions employed in classical statistical modeling are also used in statistical learning theory when comparing probability distributions. However, other loss functions are also considered for different kinds of modeling problems.

In general, statistical learning theory considers the loss $Q(\mathbf{z}, \alpha)$ between a data vector $\mathbf{z}$ and a specific model α. In the case of a parametric family of models, the notation introduced earlier is extended so that α defines both the specific parameters of the model and the parametric family to which the model belongs. In this way, models from different families can be compared. When modeling the joint probability density of the data, the appropriate loss function is the same joint negative log-likelihood used in classical statistical modeling:

$$Q(\mathbf{z}, \alpha) = -\log p(\mathbf{z}; \alpha) \ .$$

Similarly, when the data vector $\mathbf{z}$ can be decomposed into two components, $\mathbf{z} = [\mathbf{x}, y]$ and we are interested in modeling the conditional probability distribution of y as a function of $\mathbf{x}$, then the conditional negative log likelihood is the appropriate loss function:

$$Q(\mathbf{z}, \alpha) = -\log p(y \mid \mathbf{x}; \alpha) \ .$$

On the other hand, if we are not interested in the actual distribution of y but only in constructing a predictor $f(\mathbf{x}; \alpha)$ for y that minimizes the probability of making an incorrect prediction, then the 0/1 loss function used in pattern recognition is appropriate:

$$Q(\mathbf{z}, \alpha) = \begin{cases} 0, & \text{if } f(\mathbf{x}; \alpha) = y, \\ 1, & \text{if } f(\mathbf{x}; \alpha) \neq y. \end{cases}$$

In general, $Q(\mathbf{z}, \alpha)$ can be chosen depending on the nature of the modeling problem one faces. Its purpose is to measure the performance of a model so that the best model can be selected. The only requirement from the point of view of statistical learning theory is that, by convention, smaller losses imply better models of the data.

Once a loss function has been selected, identifying the best model would be relatively easy if we already knew all of the statistical properties of the data. If the data vector $\mathbf{z}$ is generated by a random process according to the probability measure $F(\mathbf{z})$, then the best model α is the one that minimizes the expected loss $R(\alpha)$ with respect to $F(\mathbf{z})$, where

$$R(\alpha) = \int Q(\mathbf{z}, \alpha) \, dF(\mathbf{z}) \ .$$

The model that minimizes $R(\alpha)$ is optimal from a decision-theoretic point of view. In the terminology of decision theory, α is a decision vector, $\mathbf{z}$ is an outcome, and $Q(\mathbf{z}, \alpha)$ is the (negative) utility measure of the outcome given the decision. Utility measures provide a numerical encoding of which outcomes are preferred over others, as well as a quantitative measurement of the degree of uncertainty one is willing to accept in choosing a risky decision that has a low probability of obtaining a highly desirable outcome versus a more conservative decision with a high probability of a moderate outcome. Choosing the decision vector α that has the best expected (negative) utility $R(\alpha)$ produces an optimal decision consistent with the risk preferences defined by the utility measure—that is, the best model given the loss function.

Unfortunately, in practice, the expected loss $R(\alpha)$ cannot be calculated directly because the probability measure $F(\mathbf{z})$ that defines the statistical properties of the data is unknown. Instead, one must choose the most suitable model one can identify based on a set of observed data vectors $\mathbf{z}_i$, $i=1,\ldots,\ell$. Assuming that the observed vectors are statistically independent and identically distributed, the average loss $R_{\mathrm{emp}}(\alpha,\ell)$ for the observed data can be used as an empirical estimator of the expected loss, where

$$R_{\mathrm{emp}}(\alpha,\ell) = \frac{1}{\ell}\sum_{i=1}^{\ell} Q(\mathbf{z}_i,\alpha) \ .$$

Statistical learning theory presumes that models α are chosen by minimizing $R_{\mathrm{emp}}(\alpha,\ell)$. Note that this presumption is consistent with standard model-fitting procedures used in statistics in which models and/or their parameters are selected by optimizing numerical criteria of this general form.

The fundamental question of statistical learning theory is the following: under what conditions does minimizing the average empirical loss $R_{\mathrm{emp}}(\alpha,\ell)$ yield models that also minimize the expected loss $R(\alpha)$, since the latter is what we actually want to accomplish? This question is answered by considering the accuracy of the empirical loss estimate. As in classical statistics, accuracy is expressed in terms of confidence regions; however, in this case, confidence regions are constructed for the expected losses, not for the parameters. The expected loss $R(\alpha)$ for a model α is regarded as fixed but unknown, since the probability measure $F(\mathbf{z})$ that defines the statistical properties of the data vectors is fixed but unknown. On the other hand, the average empirical loss $R_{\mathrm{emp}}(\alpha,\ell)$ is a random quantity that we can sample, since its value depends on the values of the observed data vectors $\mathbf{z}_i$, $i=1,\ldots,\ell$, used in its calculation. Statistical learning theory therefore considers confidence regions for $R(\alpha)$ given $R_{\mathrm{emp}}(\alpha,\ell)$.

To construct these confidence regions, we need to consider the probability distribution of the difference between the expected and average empirical losses *while taking into account the fact that models are selected by minimizing average empirical loss*. This latter caveat is the key issue that distinguishes statistical learning theory from classical statistics. One of the fundamental theorems of statistical learning theory shows that, in order to account for the fact that models are selected by minimizing average empirical loss, one must consider the maximum difference between the expected and average empirical losses; that is, one must consider the distribution of

$$\sup_{\alpha \in \mathbf{\Lambda}} \left| R(\alpha) - R_{\text{emp}}(\alpha, \ell) \right| ,$$

where $\mathbf{\Lambda}$ is the set of models one is selecting from.

The reason that the maximum difference must be considered has to do with the phenomenon of overfitting. Intuitively speaking, overfitting occurs when the set of models to choose from has so many degrees of freedom that one can find a model that fits the noise in the data but does not adequately reflect the underlying relationships. As a result, one obtains a model that looks good relative to the training data but that performs poorly when applied to new data. This mathematically corresponds to a situation in which the average empirical loss $R_{\text{emp}}(\alpha, \ell)$ substantially underestimates the expected loss $R(\alpha)$. Although there is always some probability that the average empirical loss will underestimate the expected loss for a fixed model α, both the probability and the degree of underestimation are increased by the fact that we explicitly search for the model that minimizes $R_{\text{emp}}(\alpha, \ell)$. Because of this search, the maximum difference between the expected and average empirical losses is the quantity that governs the confidence region.

The landmark contribution of Vapnik and Chervonenkis is a series of probability bounds that they have developed to construct small-sample confidence regions for the expected loss given the average empirical loss. The resulting confidence regions differ from those obtained in classical statistics in three respects. First, they do not assume that the chosen model is correct. Second, they are based on small-sample statistics and are not asymptotic approximations as is typically the case. Third, a uniform method is used to take into account the degrees of freedom in the set of models one is selecting from independent of the forms of those models. This method is based on a measurement known as the Vapnik-Chervonenkis (VC) dimension.

The VC dimension of a set of models can conceptually be thought of as the maximum number of data vectors for which one is pretty much guaranteed to find a model that fits exactly. For example, the VC dimension of a linear regression or discriminant model is equal to the number of terms in the model (i.e., the number of degrees of freedom in the classical sense), since n linear terms can be used to exactly fit n points. The actual definition of VC dimension is more general and does not formally require an exact fit; nevertheless, the intuitive insights gained by thinking about the consequences of exact fits are often valid with regard to VC dimension. For example, one consequence is that in order to avoid overfitting the number of data samples should substantially exceed the VC dimension of the set of models to choose from; otherwise, one could obtain an exact fit to arbitrary data.

Because VC dimension is defined in terms of model fitting and numbers of data points, it is equally applicable to linear, nonlinear and nonparametric models, and to combinations of dissimilar model families. This includes neural networks, classification and regression trees, classification and regression rules, radial basis functions, Bayesian networks, and virtually any other model family imaginable. In addition, VC dimension is a much better indicator of the ability of models to fit arbitrary data than is suggested by the number of parameters in the models. There are examples of models with only one parameter that have infinite VC dimension and, hence, are able to exactly fit any set of data [23, 24]. There are also models with billions of parameters that have small VC dimensions, which enables one to obtain reliable models even when the number of data samples is much less than the number of parameters. VC dimension coincides with the number of parameters only for certain model families, such as linear regression/discriminant models. VC dimension therefore offers a much more general notion of degrees of freedom than is found in classical statistics.

In the probability bounds obtained by Vapnik and Chervonenkis, the size of the confidence region is largely determined by the ratio of the VC dimension to the number of data vectors. For example, if the loss function $Q(\mathbf{z}, \alpha)$ is the 0/1 loss used in pattern recognition, then with probability at least $1 - \eta$,

$$R_{\text{emp}}(\alpha, \ell) - \frac{\sqrt{\mathcal{E}}}{2} \;\leq\; R(\alpha) \;\leq\; R_{\text{emp}}(\alpha, \ell) + \frac{\mathcal{E}}{2}\left(1 + \sqrt{1 + \frac{4\, R_{\text{emp}}(\alpha, \ell)}{\mathcal{E}}}\right) \;,$$

where

$$\mathcal{E} = \frac{4h}{\ell}\left(\ln\frac{2\ell}{h} + 1\right) - \frac{4}{\ell}\ln\left(\frac{\eta}{4}\right)$$

and where h is the VC dimension of the set of models to choose from. Note that the ratio of the VC dimension h to the number of data vectors ℓ is the dominant term in the definition of $\mathcal{E}$ and, hence, in the size of the confidence region for $R(\alpha)$. Other families of loss functions have analogous confidence regions involving the quantity $\mathcal{E}$.

The concept of VC dimension and confidence bounds for various families of loss functions are discussed in detail in books by Vapnik [22, 23, 24]. The remarkable properties of these bounds are that they make no assumptions about the probability distribution $F(\mathbf{z})$ that defines the statistical properties of the data vectors, they are valid for small sample sizes, and they are dependent only on the VC dimension of the set of models and on the properties of the loss function employed. The bounds are therefore applicable for an extremely wide range of modeling problems and for any family of models imaginable.

4.3. Model selection

As discussed at the beginning of this section, the data analyst is expected to provide not just a single parametric model, but an entire series of competing models ordered according to preference, one of which will be selected based on an examination of the data. The results of statistical learning theory are then used to select the most preferable model that best explains the data.

The selection process has two components: one is to determine a cut-off point in the preference ordering, the other is to select the model with the smallest average empirical loss $R_{\text{emp}}(\alpha, \ell)$ from among those models that occur before the cut-off. As

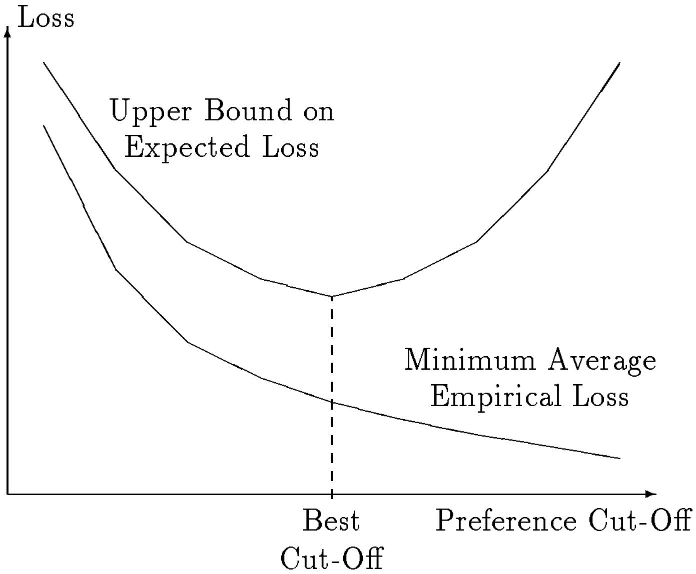

Figure 1. Expected loss and average empirical loss as a function of the preference cut-off.

the cut-off point is advanced through the preference ordering, both the set of models that appear before the cut-off and the VC dimension of this set steadily increase. This increase in VC dimension has two effects. The first effect is that with more models to choose from one can usually obtain a better fit to the data; hence, the minimum average empirical loss steadily decreases. The second effect is that the size of the confidence region for the expected loss $R(\alpha)$ steadily increases because the size is governed by the VC dimension. To choose a cut-off point in the preference ordering, Vapnik and Chervonenkis advocate minimizing the upper bound on the confidence region for the expected loss; that is, minimize the worst-case estimate of $R(\alpha)$. For example, if the 0/1 loss function were being used, one would choose the cut-off so as to minimize the left hand side of the inequality presented above for a desired setting of the confidence parameter η. The model α that minimizes the average empirical loss $R_{\mathrm{emp}}(\alpha, \ell)$ for those models that occur before the chosen cut-off is then selected as the most suitable model for the data.

The overall approach is illustrated by the graph in Figure 1. The process balances the ability to find increasingly better fits to the data against the danger of overfitting and thereby selecting a poor model. The preference ordering provides the necessary structure in which to compare competing models while at the same time taking into account their effective degrees of freedom (i.e., VC dimension). The result is a model that minimizes the worst-case loss on future data. The process itself attempts to maximize the rate of convergence to an optimum model as the quantity of available data increases.

4.4. Use of validation data

One drawback to the Vapnik-Chervonenkis approach is that it can be difficult to determine the VC dimension of a set of models, especially for the more exotic types of models. Even for simple linear regression/discriminant models, the situation is not entirely straightforward. The relationship stated above that the VC dimension is equal to the number of terms in such a model is actually an upper bound on the VC dimension. If the models are written in a certain canonical form, then the VC dimension is also bounded by the quantity $R^2A^2 + 1$, where R is the radius of the smallest sphere that encloses the available data vectors and A^2 is the sum of the squares of the coefficients of the model in its canonical form. As Vapnik has shown [23], this additional bound on the VC dimension makes it possible to obtain linear regression or discriminant models whose VC dimensions are orders of magnitude smaller than the number of terms, even if the models contain billions of terms. This fact is extremely fortunate because it offers a means of avoiding the "curse of dimensionality," enabling reliable models to be obtained even in high-dimensional spaces by basing the preference ordering of the models on the sum of the squares of the model coefficients.

In cases where the VC dimension of a set of models is difficult to determine, the expected loss can be estimated using resampling techniques [3]. In the simplest of these approaches, the available set of data is randomly divided into training and validation sets. The training set is used first to select the best-fitting model for each cut-off point in the preference ordering. The validation set is then used to estimate the expected losses of the selected models by calculating their average empirical losses on the validation data. Finally, the model with the smallest upper bound for the expected loss on the validation data is chosen as the most suitable model.

Because only a finite number of models are evaluated on the validation set (models with continuous parameters implies an infinite set of models), it is very easy to obtain confidence bounds for the expected losses of these models independent of their exact forms and without having to worry about VC dimension [23]. In particular, the same equations for the confidence bounds are used as before, except that $\mathcal{E}$ now has the value

$$\mathcal{E} = \frac{2}{\ell_v} \ln N - \frac{2}{\ell_v} \ln \eta \ ,$$

where N is the number of models evaluated against the validation set and ℓ_v is the size of the validation set. Moreover, because the number N of such models is typically small relative to the size ℓ_v of the validation set, one can obtain tight confidence regions for the expected losses of these models given their average empirical losses on the validation data. Since the same underlying principles are at work, this approach exhibits the same kind of relationship between the expected and average empirical losses as that shown in Figure 1.

Although this validation-set approach has an advantage in that it is relatively easy to obtain expected loss estimates, it has the disadvantage that dividing the available data into subsets decreases the overall accuracy of the resulting estimates. This decrease in accuracy is usually not much of a concern when data is plentiful. However, when the sample size is small, fitting models to all of the data and calculating the VC dimension for all relevant sets of models becomes more attractive.

5. Computational learning theory and PAC learning

The statistical theory of minimization of loss functions provides a general analysis of the conditions under which a class of models is learnable. The theory reduces the task of learning to that of solving a finite dimensional optimization problem: given a class of models $\aleph$, a loss function Q, and a set of data vectors $\mathbf{z}_1, \ldots, \mathbf{z}_\ell$, find the model $\alpha \in \aleph$ which minimizes the empirical loss $\sum_i Q(\mathbf{z}_i, \alpha)$.

The perfect complement to this theory would be an efficient algorithm for every class of models $\aleph$, which takes as inputs data vectors $\mathbf{z}_1, \ldots, \mathbf{z}_\ell$ and produces the model α which minimizes the empirical loss on the samples $\mathbf{z}_1, \ldots, \mathbf{z}_\ell$. Before even defining efficiency formally (we shall do so soon), we point out that such efficient algorithms are not known to exist. Furthermore, the widespread belief is that such algorithms do not exist for many classes of models. As we shall elaborate on presently, this turns out to be related to the famous question from computational mathematics: is P = NP?

Given that the answer to this question is most probably negative, the next best hope would be to characterize the model classes for which efficient algorithms do exist. Unfortunately, such characterizations are also ruled out due to the inherent undecidability of such questions. In view of these barriers, it becomes clear that the question of whether a given model class allows for an efficient algorithm to solve the minimization problem has to be tackled on an individual basis.

The computational theory of learning, initiated by Valiant's work in 1984, is devoted to the analysis of these problems. Here we describe some of the salient results. There are plenty of results that show how to solve such minimization problems for various classes of models. These show the diversity within the area of computational learning. We shall however focus on results that tend to unify the area. Thus we focus on formulating the right definition for the computational setting and on examining several parameters and attributes of the model.

5.1. Computational model of learning

The complexity of a computational task is the number of elementary steps (addition, subtraction, multiplication, division, comparison, etc.) it takes to perform the computation. This is studied as a function of the input and output size of the function to be computed. The well-entrenched and well-studied notion of efficiency is that of polynomial time: an algorithm is considered efficient if the number of elementary operations it performs is bounded by some fixed polynomial in the input and output sizes. The class of problems which can be solved by such efficient algorithms is denoted by P (for Polynomial time). This shall be our notion of efficiency as well.

In order to study the computational complexity of the learning problem, we have to define the input and output sizes carefully. The input to the learning task is a collection of vectors $\mathbf{z}_1, \ldots, \mathbf{z}_\ell \in \mathcal{R}^n$, but ℓ itself may be thought of as a parameter to be chosen by the learning algorithm. Similarly, the output of the learning algorithm is again a representation of the model, the choice of which may be left unclear by the problem. The choice could easily allow an inefficient algorithm to pass as efficient, by picking an unnecessarily large number of samples or an unnecessarily verbose representation of the hypothesis. In order to circumvent such difficulties, one forces the running time of the algorithm to be polynomial in n (the size of a single input vector) and the size of the smallest model from the class $\aleph$ that fits the data. The running time is not allowed to grow with ℓ—at least, not directly. But the smallest ℓ required to guarantee good convergence grows as a polynomial in d, the VC dimension of $\aleph$, and typically the output size of the smallest output consistent with the data will be at least d. Thus indirectly the running time is allowed to be a polynomial in ℓ.

In contrast to statistical theory, which fixes both the concept class $\aleph$ (which actually explains the data) and the hypothesis class $\aleph'$ (from which the hypothesis comes) and studies the learning problem as a function of parameters of $\aleph$ and $\aleph'$, computational learning theory usually fixes $\aleph$ and leaves the choice of $\aleph'$ to the learning algorithm. Thus $\aleph'$ is analogous to the model family $\mathbf{\Lambda}$ in statistical learning theory (Section 4.2). The only requirement from the learning algorithm is that with high probability (bounded away from 1 by a confidence parameter δ), the learning algorithm produces a hypothesis whose prediction ability is very close (given by an accuracy parameter ϵ) to the minimum loss achieved by any model from the class $\aleph$. The running time is allowed to be a polynomial in $1/\epsilon$ and $1/\delta$ as well.

The above discussion can now be formalized in the following definition, which is popularly known as the PAC model (for Probably Approximately Correct). Given a class of models $\aleph$, a loss function Q and a source of random vectors $\mathbf{z} \in \mathcal{R}^n$ that follow some unknown distribution $F(\mathbf{z})$, a *(generalized) PAC learning algorithm* is one that takes two parameters ϵ (the *accuracy* parameter) and δ (the *confidence* parameter), reads ℓ random examples $\mathbf{z}_1, \ldots, \mathbf{z}_\ell$ as input, the choice of ℓ being decided by the algorithm, and outputs a model (hypothesis) $h(\mathbf{z}_1, \ldots, \mathbf{z}_\ell)$, possibly from a class $\aleph' \neq \aleph$, such that

$$\Pr_F \left[[\mathbf{z}_1, \ldots, \mathbf{z}_\ell] \in \mathcal{R}^{n\ell} : R(h(\mathbf{z}_1, \ldots, \mathbf{z}_\ell)) \geq \inf_{\alpha \in \aleph} R(\alpha) + \epsilon \right] \leq \delta \,,$$

where $R(\,\cdot\,)$ is the same expected loss considered in statistical learning theory. The algorithm is said to be *efficient* if its running time is bounded by a polynomial in n, $1/\epsilon$, $1/\delta$ and the representation size of the α in $\aleph$ that minimizes the loss.

While the notion of generalized PAC learning (cf. [5]) is itself general enough to study any learning problem, we focus on the boolean pattern-recognition problems typically examined in computational learning theory. Here the data vector $\mathbf{z}$ is partitioned into a vector $\mathbf{x} \in \{0,1\}^{n-1}$ and a bit $y \in \{0,1\}$ that is to be predicted. The model α is given by a function $f_\alpha : \{0,1\}^{n-1} \to \{0,1\}$ and the loss function $Q(\mathbf{z}, \alpha)$ of a vector $\mathbf{z} = [\mathbf{x}, y]$ is 0 if $f_\alpha(\mathbf{x}) = y$ and 1 otherwise. In addition, the learning problem is usually noise-free in the sense that $\inf_{\alpha \in \aleph} R(\alpha) = 0$. Hence the accuracy parameter ϵ represents the maximum prediction error desired for the model.

5.2. Intractable learning problems

We focus on problems for which $Q(\mathbf{z}, \alpha)$ is computable efficiently (i.e., $f_\alpha(x)$ is computable efficiently). For such Q and $\aleph$, the problem of finding the α that minimizes Q lies in a well-studied computational class NP. NP consists of problems that can be solved efficiently by an algorithm that is allowed to make nondeterministic choices. In the case of learning, the nondeterministic machine can nondeterministically guess the α that minimizes the loss, thus solving the problem easily. Of course, the idea of an algorithm that makes nondeterministic choices is merely a mathematical abstraction, and is not efficiently realizable. The importance of the computational class NP comes from the fact that it captures many widely studied problems such as the Traveling Salesperson Problem, or the Graph Coloring Problem. Even more important is the notion of NP-hardness—a problem is NP-hard if the existence of an efficient (polynomial-time) algorithm to solve it would imply a polynomial-time algorithm to solve every problem in NP. The famous question "Is NP = P?" asks exactly this question: do NP-hard problems have efficient algorithms to solve them?

It is easy to show that several PAC learning problems are NP-hard if the hypothesis class is restricted (to something fixed). A typical example is that of learning a pattern-recognition problem: "3-term disjunctive normal form" (3-DNF). It can be shown that learning 3-term DNF formulae with 3-term DNF is NP-hard. Interestingly however it is possible to efficiently learn a broader class "3-term conjunctive normal form" which contains 3-term DNF. Thus this NP-hardness result is not pointing to any inherent computational bottlenecks to the task of learning, but merely implies that a judicious choice of the hypothesis class may be necessary to make the learning problem tractable.

It is harder to show that a class of problems is hard to learn independent of the representation of choice for the output. In order to show the hardness of such problems one needs to assume something stronger than NP $\neq$ P. A common assumption here is that there exist functions which are easy to compute, but hard to invert, even on randomly chosen instances. Such instances are common in cryptography, and in particular are the heart of well-known cryptosystems such as RSA. If this assumption is true, it implies that NP $\neq$ P. Under this assumption it is possible to show that pattern recognition problems, where the pattern is generated by a Deterministic Finite Automaton (or Hidden Markov Model) are hard to learn, under some distributions on the space of the data vectors. Recent results also show that patterns generated by constant depth boolean circuits are hard to learn under the *uniform* distribution.

In summary, the negative results shed new light on two aspects of learning. Learning is easier, i.e., more tractable, when no restrictions are placed on the model used to describe the given data. Furthermore, the complexity of the learning process is definitely dependent on the underlying distribution according to which we wish to learn.

5.3. PAC learning algorithms

We now move to some lessons learnt from positive results in learning. The first of these focuses on the role of the parameters ϵ and δ in the definition of learning. As we will see these are not very critical to the learning process. We also consider the role of "classification noise" in learning and present an alternate model which shows more robustness towards such noise.

The strength of weak learning. Of the two fuzz parameters, ϵ and δ, used in the definition of PAC learning, it seems clear that ϵ (the accuracy) is more significant than δ (the confidence), especially for pattern recognition problems. For such problems, given an algorithm which can learn a model α with probability, say 2/3 (or any confidence strictly greater than 1/2), it is easy to boost the confidence of getting a good hypothesis as follows. Pick a parameter k and run the learning algorithm k times, producing a new hypothesis each time. Denote these hypotheses by $h_1, \ldots, h_k$. Use for the new prediction the algorithm whose prediction on any vector x is the majority vote of the predictions of $h_1, \ldots, h_k$. It is easy to show, by an application of the law of large numbers, that the majority vote is ϵ-accurate with probability $1 - \exp(-ck)$ for some $c > 0$.

The accuracy parameter, on the other hand, does not appear to allow such simple boosting. It is unclear as to how one could use a learning algorithm which can learn to predict a model α with inaccuracy 1/3 to get a new algorithm which can predict a model with inaccuracy 1%. However, if we are lucky enough to be able to find learning algorithms which learn to predict with inaccuracy 1/3, independently of the distribution from which the data vectors are picked, then we could use the same learning algorithm on the region where our earlier predictions are inaccurate to boost our accuracy. Of course, the problem is that we don't know where our earlier predictions were wrong (if we knew we would change our prediction!). Though it appears that this reasoning has led us back to square one, it turns out not to be the case. In 1986, Schapire showed how to turn this intuition to get a boosting result for the accuracy parameter as well. This result demonstrates a surprising robustness of PAC learning: weak learning (with inaccuracy barely below 1/2) is equivalent to strong learning (with inaccuracy arbitrarily close to 0). However we stress that this equivalence holds only if the weak learning is independent of the data distribution. Strong learning of a model $\aleph$ under a fixed distribution F can be achieved by this method only if $\aleph$ can be learned weakly under every distribution.

Learning with noise. Most results in computational learning start by assuming that the data is observed with no noise. This is not an assumption justified by reality. It is made usually to get a basic understanding of the problem. However in order to make a computational learning result useful in practice, one must allow for noise. Numerous examples are known where an algorithm which learns without classification noise can be converted into one that can tolerate some amount of noise. However this is not universally true. To understand why some algorithms are tolerant to errors while others are not, a model of learning called the *statistical query model* has been proposed by Kearns in 1992. This model restricts a learning algorithm in the following way: instead of actually seeing data vectors $\mathbf{z}$ as sampled from the space, the learning algorithm works with an oracle and gets to ask "statistical" questions about the data vectors. A typical statistical query asks for the probability that an event defined over the data space occurs for a vector chosen at random from the distribution under which we are attempting to learn. Further, the query is presented with a tolerance parameter τ. The oracle responds with the probability of the event to within an additive error of τ. It is easy to see how to simulate this oracle, given access to random samples of the data. Furthermore, it is easy to see how to simulate this oracle even when the data vectors come with some classification noise, but less than τ. Thus learning with access only to a statistical query oracle is a sufficient condition for learning with classification noise. Almost all known algorithms that learn with classification noise can be shown to learn in the statistical query model. Thus this model provides a good standpoint from which to analyse the effectiveness of a potential learning strategy when attempting to learn in the presence of noise.

Alternate models for learning. We have focused on the PAC model since it is close to the spirit of data mining. However, a large body of work in computational learning focuses on models other than the PAC model. This body of work considers learning when one is allowed to ask questions about the data one is trying to learn. Consider for instance a handwriting recognition program, which generates some patterns and asks the teacher to indicate what letter this pattern seems to resemble. It is conceivable that such learning programs may be more efficient than passive handwriting recognition programs. A class of learning algorithms that behave in this manner has been studied under the label of learning with queries. Other models for learning that have been studied include capture scenarios of supervised learning and learning in an online setting.

5.4. Further reading on computational learning theory

We have given a very informal sketch of the new questions posed by studying the process of learning, or fitting models to given data, from the point of view of computation. Due to space limitations, we do not give a complete list of references to the sources of the results mentioned above. The interested reader is referred to the

Table 1. Statisticians' and data miners' issues in data analysis.

Statisticians' issues	Data miners' issues
Model specification	Accuracy
Parameter estimation	Generalizability
Diagnostic checks	Model complexity
Model comparison	Computational complexity
Asymptotics	Speed of computation

the text on this subject by Kearns and Vazirani [9] for detailed coverage of the topics above with complete references. Other surveys include Valiant [21] and Angluin [1]. Finally, a number of different lecture notes are now available online. We have used in particular those of Mansour [12], which include pointers to other useful home pages for tracking recent developments in computational learning and their applicability to practical scenarios.

6. Conclusions

The foregoing sections illustrate some differences of approach between classical statistics and data-mining methods that originated in computer science and engineering. Table 1 summarizes what we regard as the principal issues in data analysis that would be considered by statisticians and data miners.

In addition, the approaches of statistical learning theory and computational learning theory provide productive extensions of classical statistical inference. The inference procedures of classical statistics involve repeated sampling under a given statistical model; they allow for variation across data samples but not for the fact that in many cases the choice of model is dependent on the data. Statistical learning theory bases its inferences on repeated sampling from an unknown distribution of the data, and allows for the effect of model choice, at least within a prespecified class of models that could in practice be very large. The PAC-learning results from computational learning theory seek to identify modeling procedures that have a high probability of near-optimality over all possible distributions of the data. However, the majority of the results assume that the data are noise-free and that the target concept is deterministic. Even with these simplifications, useful positive results for near-optimal modeling are difficult to obtain, and for some modeling problems only negative results have been obtained.

To some extent, the differences between statistical and data-mining approaches to modeling and inference are related to the different kinds of problems on which these approaches have been used. For example, statisticians tend to work with relatively simple models for which issues of computational speed have rarely been a concern. Some of the differences, however, present opportunities for statisticians and data miners to learn from each other's approaches. Statisticians would do well to downplay the role of asymptotic accuracy estimates based on the assumption that the correct model has been identified, and instead give more attention to estimates of predictive accuracy obtained from data separate from those used to fit the model. Data miners can benefit by learning from statisticians' awareness of the problems caused by outliers and influential data values, and by making greater use of diagnostic statistics and plots to identify irregularities in the data and inadequacies in the model.

As noted earlier, statistical methods are particularly likely to be preferable when fairly simple models are adequate and the important variables can be identified before modeling. In problems with large data sets in which the relation between class and feature variables is complex and poorly understood, data mining methods offer a better chance of success. However, many practical problems fall between these extremes, and the variety of available models for data analysis, exemplified by those listed in Section 2.2, offers no sharp distinction between statistical and data-mining methods. No single method is likely to be obviously best for a given problem, and use of a combination of approaches offers the best chance of making secure inferences. For example, a rule-based classifier might use additional feature variables formed from linear combinations of features computed implicitly by logistic discriminant or a neural-network classifier. Inferences from several distinct families of models can be combined, either by weighting the models' predictions or by an additional stage of modeling in which predictions from different models are themselves used as input features—"stacked generalization" [30]. The overall conclusion is that statisticians and data miners can profit by studying each other's methods and using a judiciously chosen combination of them.

Acknowledgements

We are happy to acknowledge helpful discussions with several participants at the Workshop on Data Mining and its Applications, Institute of Mathematics and its Applications, Minneapolis, November 1996 (J.H.), many conversations with Vladimir Vapnik (E.P.), and comments and pointers from Yishay Mansour, Dana Ron and Ronitt Rubinfeld (M.S.).

References

[1] Angluin, D. (1992). Computational learning theory: survey and selected bibliography. In *Proceedings of the Twenty Fourth Annual Symposium on Theory of Computing*, 351–369. ACM.

[2] Cox, D. R. and Hinkley, D. V. (1986). *Theoretical statistics.* London: Chapman and Hall.

[3] Efron, B. (1981). *The jackknife, the bootstrap, and other resampling plans*, CBMS Monograph 38. Philadelphia, Pa.: SIAM.

[4] Hand, D. J. (1981). *Discrimination and classification.* Chichester, U.K.: Wiley.

[5] Haussler, D. (1990). Decision theoretic generalizations of the PAC learning model. In *Algorithmic Learning Theory*, eds. S. Arikawa, S. Goto, S. Ohsuga, and T. Yokomori, pp. 21–41. New York: Springer-Verlag.

[6] Huber, P. J. (1981). *Robust statistics.* New York: Wiley.

[7] John, G., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference*, pp. 121–129. San Mateo, Calif.: Morgan Kaufmann.

[8] Journel, A. G., and Huibregts, C. J. (1978). *Mining geostatistics.* London: Academic Press.

[9] Kearns, M. J., and Vazirani, U. V. (1994). *An introduction to computational learning theory.* Cambridge, Mass.: MIT Press.

[10] Kononenko, I., and Hong, S. J. (1997). Attribute selection for modeling. *Future Generation Computer Systems*, this issue.

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

[12] Mansour, Y. Lecture notes on learning theory. Available from `http://www.math.tau.ac.il/~mansour`.

[13] Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (eds.) (1994). *Machine learning, neural and statistical classification.* Hemel Hempstead, U.K.: Ellis Horwood.

[14] Miller, A. J. (1983). Contribution to the discussion of "Regression, prediction and shrinkage" by J. B. Copas. *Journal of the Royal Statistical Society, Series B*, **45**, 346–347.

[15] Pearl, J. (1995). Causal diagrams for empirical research. *Biometrika*, **82**, 669–710.

[16] Ripley, B. D. (1994). Comment on "Neural networks: a review from a statistical perspective" by B. Cheng and D. M. Titterington. *Statistical Science*, **9**, 45–48.

[17] Rissanen, J. (1989). *Stochastic Complexity in Statistical Inquiry.* River Edge, N.J.: World Scientific.

[18] Sakamoto, Y., Ishiguro, M., and Kitagawa, G. (1986). *Akaike information criterion statistics.* Dordrecht, Holland: Reidel.

[19] Tukey, J. W. (1977). *Exploratory data analysis.* Reading, Mass.: Addison-Wesley.

[20] Vach, W., Rossner, R., and Schumacher, M. (1996). Neural networks and logistic regression: part II. *Computational Statistics and Data Analysis*, **21**, 683–701.

[21] Valiant, L. (1991). A view of computational learning theory. In *Computation and Cognition: Proceedings of the First NEC Research Symposium*, 32–51. Philadelphia, Pa.: SIAM.

[22] Vapnik, V. N. (1982). *Estimation of dependencies based on empirical data.* New York: Springer-Verlag.

[23] Vapnik, V. N. (1995). *The nature of statistical learning theory.* New York: Springer-Verlag.

[24] Vapnik, V. N. (to appear, 1997). *Statistical learning theory.* New York: Wiley.

[25] Vapnik, V. N., and Chervonenkis, A. Ja. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, **16**, 264–280. Originally published in *Doklady Akademii Nauk USSR*, **181** (1968).

[26] Vapnik, V. N., and Chervonenkis, A. Ja. (1981). Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theory of Probability and its Applications*, **26**, 532–553.

[27] Vapnik, V. N., and Chervonenkis, A. Ja. (1991). The necessary and sufficient conditions for consistency of the method of empirical risk minimization. *Pattern Recognition and Image Analysis*, **1**, 284–305. Originally published in *Yearbook of the Academy of Sciences of the USSR on Recognition, Classification, and Forecasting*, **2** (1989).

[28] Wallace, C. S. and Freeman, P. R. (1987). Estimation and inference by compact codes. *Journal of the Royal Statistical Society, Series B,* **49**, 240–252.

[29] Weisberg, S. (1985). *Applied regression analysis*, 2nd edn. New York: Wiley.

[30] Wolpert, D. (1992). Stacked generalization. *Neural Networks*, **5**, 241–259.