Model Selection and the Principle of Minimum Description Length

Mark H. HANSEN and Bin YU

This article reviews the principle of minimum description length (MDL) for problems of model selection. By viewing statistical modeling as a means of generating *descriptions* of observed data, the MDL framework discriminates between competing models based on the *complexity* of each description. This approach began with Kolmogorov's theory of algorithmic complexity, matured in the literature on information theory, and has recently received renewed attention within the statistics community. Here we review both the practical and the theoretical aspects of MDL as a tool for model selection, emphasizing the rich connections between information theory and statistics. At the boundary between these two disciplines we find many interesting interpretations of popular frequentist and Bayesian procedures. As we show, MDL provides an objective umbrella under which rather disparate approaches to statistical modeling can coexist and be compared. We illustrate the MDL principle by considering problems in regression, nonparametric curve estimation, cluster analysis, and time series analysis. Because model selection in linear regression is an extremely common problem that arises in many applications, we present detailed derivations of several MDL criteria in this context and discuss their properties through a number of examples. Our emphasis is on the practical application of MDL, and hence we make extensive use of real datasets. In writing this review, we tried to make the descriptive philosophy of MDL natural to a statistics audience by examining classical problems in model selection. In the engineering literature, however, MDL is being applied to ever more exotic modeling situations. As a principle for statistical modeling in general, one strength of MDL is that it can be intuitively extended to provide useful tools for new problems.

KEY WORDS: AIC; Bayesian methods; Bayes information criterion; Cluster analysis; Code length; Coding redundancy; Information theory; Model selection; Pointwise and minimax lower bounds; Regression; time series.

1. OVERVIEW

The principle of parsimony, or Occam's razor, implicitly motivates the process of data analysis and statistical modeling and is the soul of model selection. Formally, the need for model selection arises when investigators must decide among model classes based on data. These classes might be indistinguishable from the standpoint of existing subject knowledge or scientific theory, and the selection of a particular model class implies the confirmation or revision of a given theory. To implement the parsimony principle, one must quantify "parsimony" of a model relative to the available data. Applying this measure to a number of candidates, we search for a concise model that provides a good fit to the data. Rissanen (1978) distilled such thinking in his principle of minimum description length (MDL): Choose the model that gives the shortest description of data. In this framework a concise model is one that is easy to describe, whereas a good fit implies that the model captures or describes the important features evident in the data.

MDL has its intellectual roots in the algorithmic or descriptive complexity theory of Kolmogorov, Chaitin, and Solomonoff (cf. Li and Vitányi 1996). Kolmogorov, the founder of axiomatic probability theory, examined the relationship between mathematical formulations of randomness and their application to real-world phenomena. He ultimately turned to algorithmic complexity as an alternative means of expressing random events. A new characterization of probability emerged based on the length of the shortest binary computer program that describes an object or event. (A program can "describe" an object by "printing" or in some way exhibiting the object. Typically, an object is a binary string, and exhibiting the string is nothing more than printing the individual 0's and 1's in order and stopping in finite time.) We refer to this quantity as the descriptive complexity of the object. Up to a constant, it can be defined independent of any specific computing device, making it a universal quantity Kolmogorov 1965, 1968 (cf., Cover and Thomas 1991). Because this descriptive complexity is universal, it provides a useful way to think about probability and other problems that build on fundamental notions of probability. In theory, it can also be used to define inductive inference in general (or statistical inference in particular) as the search for the shortest program for data.

Unfortunately, the descriptive complexity of Kolmogorov is not computable (cf. Cover and Thomas 1991) and thus cannot be used as a basis for inference given real data. Rissanen modified this concept when proposing MDL, sidestepping computability issues. First, he restricted attention to only those descriptions that correspond to probability models or distributions (in the traditional sense); he then opted to emphasize the description length interpretation of these distributions rather than the actual finite-precision computations involved. In so doing, Rissanen derived a broad but usable principle for statistical modeling. By considering only probability distributions as a basis for generating descriptions, Rissanen endowed MDL with a rich information-theoretic interpretation; description length can be thought of as the number of digits in a binary string used to code the data for transmission. Formally, then, he equated the task of "describing" data with coding. Not surprisingly, the development of MDL borrowed heavily from Shannon's work on coding theory (Shannon 1948). Because of the close ties, we frequently use the terms "code length"

Mark H. Hansen is Member of the Technical Staff, Statistics and Data Mining Research Department of Bell Laboratories in Murray Hill, NJ. Bin Yu is Associate Professor in statistics at University of California in Berkeley. Her research was partially supported by NSF grants DF98-02314 and DMS-9803063, and ARO grant DAAG55-98-1-0341. The authors thank Jianhua Huang for his help with a preliminary draft of this article. The authors would also like to thank Ed George, Robert Kohn, Wim Sweldens, Martin Wells, Andrew Gelman, John Chambers and two anonymous referees for helpful comments.

Journal of the American Statistical Association
June 2001, Vol. 96, No. 454, Review Paper

and "description length" interchangeably. As we demonstrate, the connection between MDL and information theory provides new insights into familiar statistical procedures.

In Rissanen's formulation of MDL, any probability distribution is considered from a descriptive standpoint; that is, it is not necessarily the underlying data-generating mechanism (although it does not exclude such a possibility). Thus MDL extends the more traditional random sampling approach to modeling. Many probability distributions can be compared in terms of their descriptive power, and if the data in fact follow one of these models, then Shannon's celebrated source coding theorem (cf. Cover and Thomas 1991) states that this "true" distribution gives the MDL of the data (on average and asymptotically).

An important precursor to Rissanen's MDL is the work of Wallace and Boulton (1968), who applied the idea of minimum message length (MML) to clustering problems. While based on code length, MML exclusively uses a two-part coding formulation that is most natural in parametric families (see Sec. 4.2; Baxter and Oliver 1995; Wallace and Freeman 1987). The original MML proposal stopped short of a framework for addressing other modeling problems, and recent advances seem to focus mainly on parameter estimation. In contrast, Rissanen formulated MDL as a broad principle governing statistical modeling in general. Two other approaches to model selection that are influential and important in their own right are those of Akaike (1974) and Schwarz (1978). In his derivation of *AIC*, A Information Criterion, Akaike (1974) gives for the first time formal recipes for general model selection problems from the point of view of prediction. It is fascinating to note the crucial role that the information-theoretic Kullback–Leibler divergence played in the derivation of AIC, because we demonstrate in this article that Kullback–Leibler divergence is indispensable in the MDL framework. Schwarz (1978) took a Bayesian approach to model selection, deriving an approximation to a Bayesian posterior when the posterior exists. This approximate Bayesian model selection criterion has a form very similar to AIC and is termed the Bayesian information criterion (BIC).

MDL has connections to both frequentist and Bayesian approaches to statistics. If we view statistical estimation in a parametric family as selecting models (or distributions) indexed by the parameters, then MDL gives rise to the maximum likelihood (ML) principle of parameter estimation in classical statistics. It is therefore a generalization of the ML principle to model selection problems where ML is known to fail. The performance of MDL criteria has been evaluated very favorably based on the random sampling or frequentist paradigm (e.g., Barron, Rissanen, and Yu 1998; Hannan, McDougall, and Poskitt 1989; Hannan and Rissanen 1982; Lai and Lee 1997; Speed and Yu 1994; Wei, 1992). Moreover, MDL has close ties with the Bayesian approach to statistics. For example, BIC has a natural interpretation in the MDL paradigm, and some forms of MDL coincide with Bayesian schemes (cf. Sec. 3). Because of the descriptive philosophy, the MDL paradigm serves as an objective platform from which we can compare Bayesian and non-Bayesian procedures alike.

The rest of the article is organized as follows. Section 2 introduces basic coding concepts and explains the MDL principle. We begin with Kraft's inequality, which establishes the equivalence between probability distributions and code lengths. We illustrate different coding ideas through a simple example of coding or compressing up-and-down indicators derived from daily statistics of the Dow Jones industrial average. We emphasize that using a probability distribution for coding or description purposes does not require that it actually generate our data. We revisit MDL at the end of Section 2 to connect it to the ML principle and Bayesian statistics. We also define the notion of a "valid" description length, in the sense that valid coding schemes give rise to MDL selection rules that have provably good performance. (This issue is explored in depth in Sec. 5.) Section 3 formally introduces different forms of MDL such as two-stage (or multistage in general), mixture, predictive, and normalized ML.

Section 4 contains applications of MDL model selection criteria in linear regression models, curve estimation, cluster analysis, and time series models. Our coverage on regression models is extensive. We compare well-known MDL criteria to AIC and BIC through simulations and real applications. These studies suggest an adaptive property of some forms of MDL, allowing them to behave like AIC or BIC, depending on which is more desirable in the given context. (Hansen and Yu 1999 further explored this property.) Cluster analysis is also considered in Section 4, where we apply MML (Wallace and Boulton 1968). We end this section by fitting an autoregressive moving average (ARMA) model to the Dow Jones datasets, comparing predictive MDL (PMDL), AIC and BIC for order selection.

Section 5 reviews theoretical results on MDL. These are the basis or justification for different forms of MDL to be used in parametric model selection. In particular, we mention the remarkable pointwise lower bound of Rissanen (1986a) on expected (coding) redundancy and its minimax counterpart of Clarke and Barron (1990). Both lower bounds are extensions of Shannon's source coding theorem to universal coding. Section 5 ends with an analysis of the consistency and prediction error properties of MDL criteria in a simple example.

2. BASIC CODING CONCEPTS AND THE MINIMUM DESCRIPTION LENGTH PRINCIPLE

2.1 Probability and Idealized Code Length

2.1.1 The Discrete Case. A *code* $\mathcal{C}$ on a set $\mathcal{A}$ is simply a mapping from $\mathcal{A}$ to a set of *codewords*. In this section we consider binary codes so that each codeword is a string of 0's and 1's. Let $\mathcal{A}$ be a finite set and let Q denote a probability distribution on $\mathcal{A}$. The fundamental premise of the MDL paradigm is that $-\log_2 Q$, the negative logarithm of Q, can be viewed as the code length of a binary code for elements or symbols in $\mathcal{A}$.

Example 1: Huffman's Algorithm. Let $\mathcal{A} = \{a, b, c\}$ and let Q denote a probability distribution on $\mathcal{A}$ with $Q(a) = 1/2$ and $Q(b) = Q(c) = 1/4$. Following Huffman's algorithm (Cover and Thomas 1991, p. 92), we can construct a code for $\mathcal{A}$ by growing a binary tree from the end nodes $\{a, b, c\}$. This procedure is similar to the greedy algorithm used in agglomerative, hierarchical clustering (Jobson 1992). First, we choose the two elements with the smallest probabilities, b and

c, and connect them with leaves 0 and 1, assigned arbitrarily, to form the intermediate node bc with node probability $1/4+1/4=1/2$. We then iterate the process with the new set of nodes $\{a, bc\}$. Because there are only two nodes left, we connect a and bc with leaves 0 and 1, again assigned arbitrarily, and reach the tree's root. The tree obtained through this construction, along with the resulting code, are given explicitly in Figure 1. Let L be the code length function associated with this code so that $L(a)=L(0)=1$, $L(b)=L(10)=2$, and $L(c)=L(11)=2$. It is easy to see that in this case, our code length is given exactly by $L(x)=-\log_2 Q(x)$ for all $x\in\mathcal{A}$. When we encounter ties in this process, Huffman's algorithm can produce different codes depending on how we choose which nodes to merge. For example, suppose that we start with a uniform distribution on $\mathcal{A}$, $Q(a)=Q(b)=Q(c)=1/3$. At the first step in Huffman's algorithm, if we join a and b, then the resulting code is $a\to 00$, $b\to 01$, and $c\to 1$. On the other hand, if we begin by joining b and c, then we arrive at the same code as in Figure 1. Fortunately, no matter how we handle ties, the expected length (under Q) of the resulting code is always the same; that is, the expected value of $L(x)$ computed under the distribution $Q(x)$ will be the same for all Huffman codes computed for $Q(x)$.

Clearly, the Huffman code constructed in our example is not unique, because we can permute the labels at each level in the tree. Moreover, depending on how we settle ties between the merged probabilities at each step in the algorithm, we can obtain different codes with possibly different lengths. We illustrated this point in the example, where we also indicated that despite these differences, the expected length of the Huffman code (under the distribution Q) is always the same. An interesting feature of the code in Example 1 is that any string of 0's and 1's can be uniquely decoded without introducing separating symbols between the codewords. The string 0001110, for example, must have come from the sequence $aaacb$. Given an arbitrary code, if no codeword is the prefix of any other, then unique decodability is guaranteed. Any code satisfying this codeword condition is called a *prefix code*. By taking their codewords as end nodes of a binary tree, all Huffman codes are in this class.

In general, there is a correspondence between the length of a prefix code and the quantity $-\log_2 Q$ for a probability distribution Q on $\mathcal{A}$. An integer-valued function L corresponds to the code length of a binary prefix code if and only if it

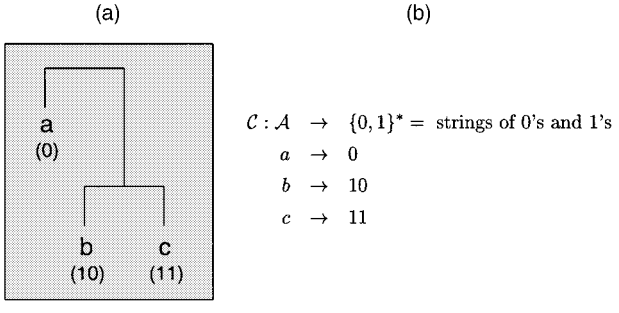

Figure 1. Constructing a Huffman code in Example 1. (a) The binary tree on which the code is based; (b) the final mapping.

satisfies Kraft's inequality,

$$\sum_{x\in\mathcal{A}} 2^{-L(x)} \le 1 \tag{1}$$

(see Cover and Thomas 1991 for a proof). Therefore, given a prefix code $\mathcal{C}$ on $\mathcal{A}$ with length function L, we can define a distribution on $\mathcal{A}$ as

$$Q(x)=\frac{2^{-L(x)}}{\sum_{z\in\mathcal{A}} 2^{-L(z)}} \quad \text{for any } x\in\mathcal{A}.$$

Conversely, for any distribution Q on $\mathcal{A}$ and any $x\in\mathcal{A}$, we can find a prefix code with length function $L(x)=\lceil -\log_2 Q(x)\rceil$, the smallest integer greater than or equal to $-\log_2 Q(x)$. Despite our good fortune in Example 1, Huffman's algorithm does not necessarily construct a code with this property for every distribution Q. [We can only guarantee that the length function L derived from Huffman's algorithm is within 2 of $\lceil -\log_2 Q\rceil$. Although slightly more complicated, the Shannon–Fano–Elias coder produces a length function that satisfies $L=\lceil -\log_2 Q\rceil$ exactly (Cover and Thomas 1991).]

Now, suppose that elements or symbols of $\mathcal{A}$ are generated according a known distribution P or, in statistical terms, that we observe data drawn from P. Given a code $\mathcal{C}$ on $\mathcal{A}$ with length function L, the *expected code length* of $\mathcal{C}$ with respect to P is defined as

$$L_{\mathcal{C}}=\sum_{x\in\mathcal{A}} P(x)L(x). \tag{2}$$

As we have seen, if $\mathcal{C}$ is a prefix code, then L is essentially equivalent to $-\log_2 Q$ for some distribution Q on $\mathcal{A}$. Shannon's Source Coding Theorem states that the expected code length (2) is minimized when $Q=P$, the true distribution of our data.

Theorem 1: Shannon's Source Coding Theorem. Suppose that elements of $\mathcal{A}$ are generated according to a probability distribution P. For any prefix code $\mathcal{C}$ on $\mathcal{A}$ with length function L, the expected code length $L_{\mathcal{C}}$ is bounded below by $H(P)$, the entropy of P. That is,

$$L_{\mathcal{C}}\ge H(P)\equiv -\sum_{a\in\mathcal{A}} P(a)\log_2 P(a), \tag{3}$$

where equality holds if and only if $L=-\log_2 P$.

The proof of the "if" part of this theorem follows from Jensen's inequality, and the "only if" part is trivial. Broadly, codes based on P remove redundancy from the data without any loss of information by assigning short codewords to common symbols and long codewords to rare symbols. (We provide a formal definition of redundancy in Section 5.) This is the same rationale behind Morse code in telegraphy.

By applying Huffman's algorithm to the distribution P, we obtain a code that is nearly optimal in expected code length. Cover and Thomas (1991) prove that the Huffman code for P has an expected length no greater than $H(P)+1$. We must emphasize, however, that any distribution Q defined on $\mathcal{A}$ — not necessarily the data-generating or true distribution P — can be used to encode data from $\mathcal{A}$. In most statistical applications, the true distribution P is rarely known, and to a large

extent this article is concerned with codes built from various approximations to P.

Ultimately, the crucial aspect of the MDL framework is not found in the specifics of a given coding algorithm, but rather in the code length interpretation of probability distributions. For simplicity, we refer to $L_Q = -\log_2 Q$ as the *code length* of (the code corresponding to) a distribution Q, whether or not it is an integer. The unit is a *bit*, which stands for *bi*nary digi*t* and is attributed to John W. Tukey. (Later in the article, we also use the unit *nat* when a natural logarithm is taken.)

Example 2: Code Length for Finitely Many Integers. Consider the finite collection of integers $\mathcal{A} = \{1, 2, 3, \ldots, N\}$, and let Q denote the uniform distribution on $\mathcal{A}$, so that $Q(k) = 1/N$ for all $k \in \mathcal{A}$. Let $\lfloor \log_2 N$ be the integer part of $\log_2 N$. By applying Huffman's algorithm in this setting, we obtain a uniform code with length function that is not greater than $\lfloor \log_2 N$ for all k but is equal to $\lfloor \log_2 N$ for at least two values of k. Whereas we know from Shannon's Source Coding Theorem that an expected code length of such a code is optimal only for a true uniform distribution, this code is a reasonable choice when very little is known about how the data were generated. This is simply a restatement of Laplace's Principle of Indifference, which is often quoted to justify the assignment of uniform priors for a Bayesian analysis in discrete problems.

Example 3: Code Length for Natural Numbers. Elias (1975) and Rissanen (1983) constructed a code for the natural numbers $\mathcal{A} = \{1, 2, 3, \ldots\}$ starting with the property that the code length function decreases with $a \in \mathcal{A}$. The rate of decay is then taken to be as small as possible, subject to the constraint that the length function must still satisfy Kraft's inequality. Rissanen argued that the resulting prefix code is "universal" in the sense that it achieves essentially the shortest coding of large natural numbers. Its length function is given by

$$\log_2^* n := \sum_{j>1} \max(\log_2^{(j)} n, 0) + \log_2 c_0, \tag{4}$$

where $\log_2^{(j)}(\cdot)$ is the jth composition of $\log_2$, (e.g., $\log_2^{(2)} n = \log_2 \log_2 n$) and

$$c_0 := \sum_{n>1} 2^{-\log_2^* n} = 2.865\ldots$$

2.1.2 The Continuous Case. Suppose that our data are no longer restricted to a finite set, but instead range over an arbitrary subset of the real line. Let f denote the data-generating or true density. Given another density q defined on $\mathcal{A}$, we can construct a code for our data by first discretizing $\mathcal{A}$ and then applying, say, Huffman's algorithm. In most statistical applications, we are not interested in $\mathcal{A}$, but rather in its Cartesian product $\mathcal{A}^n$ corresponding to an n-dimensional continuous data sequence $x^n = (x_1, \ldots, x_n)$. Then, if we discretize $\mathcal{A}$ into equal cells of size δ, the quantity $-\log_2(q(x^n) \times \delta^n) = -\log_2 q(x^n) - n \log_2 \delta$ can be viewed as the code length of a prefix code for the data sequence x^n. We say that δ is the precision of the discretization, and for fixed δ we refer to $-\log_2 q(x^n)$ as an *idealized code length.* In Section 3.1 we return to discretization issues arising in modeling problems.

From a straightforward generalization of Shannon's source coding theorem to continuous random variables, it follows that the best code for a data string x^n is based on its true or generating density $f(x^n)$. In this case, the lower bound on the expected code length is the differential entropy

$$H(f) = -\int \log_2 f(x^n) f(x^n)\, dx^n. \tag{5}$$

2.2 A Simple Example

In this section we consider coding a pair of long binary strings. We not only illustrate several different coding schemes, but also explore the role of postulated probability models Q in building good codes. This is a valuable exercise, whether or not it is appropriate to believe that these strings are actually generated by a specific probabilistic mechanism. Although our emphasis is on coding for compression purposes, we have framed the following example so as to highlight the natural connection between code length considerations and statistical model selection. Each of the coding schemes introduced here is discussed at length in the next section, where we take up modeling issues in greater detail.

Example 4: Code Length for Finite, Binary Strings. For the 6,430-day trading period July 1962–June 1988, we consider two time series derived from the Dow Jones industrial average (DJIA). Let p_t denote the logarithm of the index at day t and define the daily return, R_t, and the intraday volatility, V_t, as

$$R_t = P_t - P_{t-1} \qquad \text{and} \qquad V_t = .9V_{t-1} + .1R_t^2, \tag{6}$$

where V_0 is the unconditional variance of the series P_t. The data for this example were taken from the Social Sciences Data Collection at UC San Diego (SSDC, 2001) where one can also find references for the definitions (6).

Consider two "up-and-down" indicators derived from the daily return and intraday volatility series. The first indicator takes the value 1 if the return R_t on a given day was higher than that for the previous day, R_{t-1} (an "up"), and 0 otherwise (a "down"). In terms of the original (logged) DJIA series P_t, we assign the value 1 if $P_t - 2P_{t-1} + P_{t-2} \geq 0$, so that our first indicator is derived from a moving average process. The second variable is defined similarly, but instead tracks the volatility series, making it a function of another moving average process. This gives us two binary strings of length $n = 6{,}430 - 1 = 6{,}429$. There are 3,181 or 49.49% 1's or ups in the return difference indicator string, compared to 2,023 or 31.47% 1's in the volatility difference string. Figure 2 presents the last 1,000 observations from each series. To coordinate with our construction of binary strings, we have plotted daily differences so that ups correspond to positive values and downs correspond negative values. In the panels below these plots, grayscale maps represent the average number of ups calculated in 10-day intervals (with black representing 10 consecutive trading days for which the given series increased and white indicating a period of 10 downs). The activity clearly evident at the right in these plots corresponds to the stock market crash of October 19, 1987. As one might expect, the intraday volatility jumped dramatically, whereas the overall return was down sharply from the previous day.

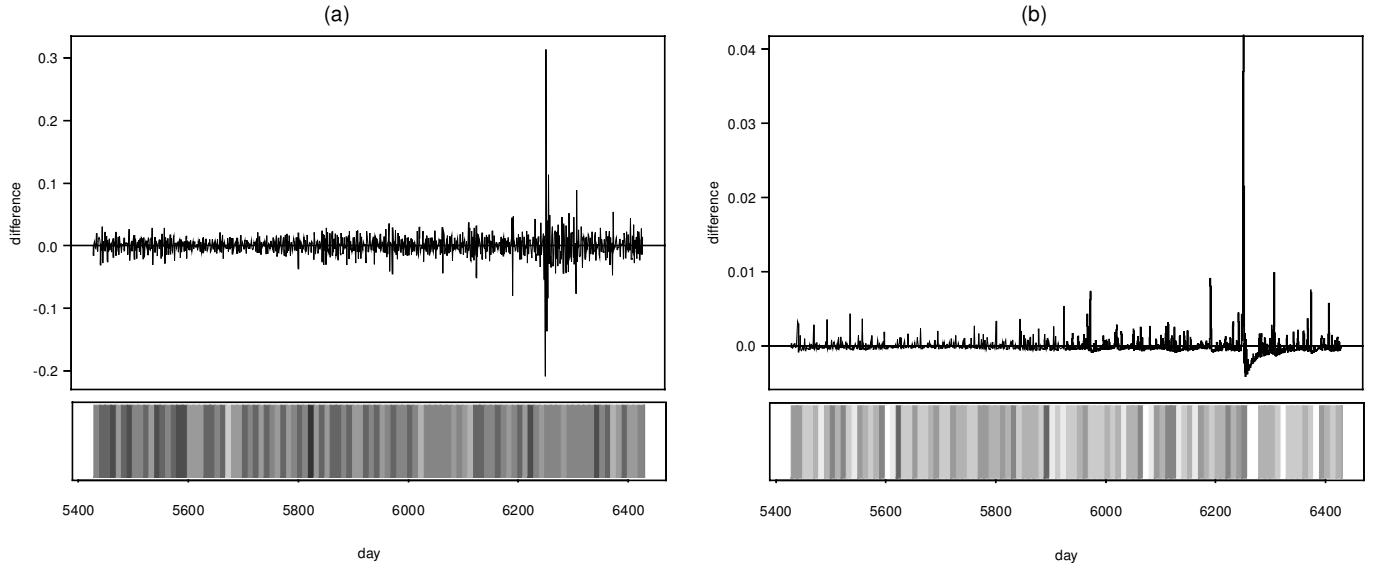

Figure 2. Differences of the Daily Returns (a) and Volatility (b) Series for the Last 1,000 Days of Our Study Period. The horizontal line in each plot corresponds to $y = 0$. The grayscale maps represent the average number of ups calculated in 10-day intervals (with black representing 10 consecutive trading days for which the given series increased; and white indicating a period of 10 downs).

Using these strings, we describe three coding algorithms, each assuming that the length of the string, $n = 6{,}429$, is known to both sender and receiver. Imagine, for example, that a financial firm in San Francisco needs to transmit this up-and-down information (as a batch) to its branch in San Diego. Clearly, each string can be transmitted directly without any further coding, requiring $n = 6{,}429$ bits. By entertaining different probability distributions, however, we might be able to decrease the code length needed to communicate these sequences.

Two-Stage Coding. Suppose that the sender uses a Bernoulli(p) model to send the series. Then p must be estimated from the series and sent first. Let k be the number of ups in the series, so that there are only n different $p = k/n$'s that one could send. Using the uniform coding scheme of Example 2, this takes $\log_2 n = 6{,}429$, or 13 bits. Once p is known to both sender and receiver it can be used in the next stage of coding. For example, suppose that we view a string $x^n = (x_1, \ldots, x_n) \in \{0, 1\}^n$ as n iid observations from the Bernoulli distribution with $p = k/n$. From the form of this distribution, it is easy to see that we can encode every symbol in the string at a cost of $-\log_2(k/n)$ bits for a 1 and $-\log_2(1 - k/n)$ bits for a 0. Therefore, transmitting each sequence requires an additional $-k\log_2(k/n) - (n - k)\log_2(1 - k/n)$ bits after p is known, giving us a total code length of

$$\log_2 n + [-k\log_2(k/n) - (n-k)\log_2(1-k/n)]. \qquad (7)$$

Under this scheme, we pay 6,441 ($> 6{,}429$) bits to encode the ups and downs of the return series, but only 5,789 ($< 6{,}429$) bits for the volatility series. Therefore, relative to sending this information directly, we incur an extra cost of .2% on the return string, but save 10% on the volatility string.

From a modeling standpoint, we could say that an iid Bernoulli model is postulated for compression or coding of a given string and that the Bernoulli probability p is estimated by k/n. The first term in (7) is the code length for sending k or the estimated p, whereas the second term is the code length for transmitting the actual string using the Bernoulli model or encoder. The success of the probability model is determined by whether there is a reduction in code length relative to the n bits required without a model. From the second term in (7), we expect some improvement provided that k/n is not too close to $1/2$, and this saving should increase with n. But when $k = n/2$,

$$-k\log_2(k/n) - (n-k)\log_2(1-k/n) = n,$$

and the Bernoulli model does not help. Considering our daily up-and-down information, we were able to decrease the code length for transmitting the volatility string by about 10%, because the proportion of 1's in this sequence is only .31. For the return string, on the other hand, the proportion of ups is close to $1/2$, so that the second term in (7) is 6,428, just 1 bit shy of $n = 6{,}429$. After adding the additional 13-bit cost to transmit p, the Bernoulli encoder is outperformed by the simple listing of 0's and 1's.

Mixture Coding (With a Uniform Prior). If we assume that each binary string comprises iid observations, then by independence we obtain a joint distribution on x^n that can be used to construct a coder for our daily up-and-down information. Suppose, for example, that we postulate an iid Bernoulli model, but rather than estimate p, we assign it a uniform prior density on $[0, 1]$. We can then apply the resulting mixture distribution to encode arbitrary binary strings. If, for example, a sequence $x^n = (x_1, \ldots, x_n)$ consists of k 1's and $(n - k)$ 0's, then

$$m(x^n) = \int_0^1 p^k(1-p)^{n-k}\,dp = \frac{\Gamma(k+1)\Gamma(n-k+1)}{\Gamma(n+2)} = \frac{k!(n-k)!}{(n+1)!},$$

where m is used to denote a "mixture." Thus the code length of this (uniform) mixture code is

$$-\log_2 m(x^n) = -\log_2 k!(n-k)! + \log_2(n+1)!. \quad (8)$$

In terms of our original binary series, by using this mixture code we incur a cost of 6,434 bits to transmit the return string and 5,782 bits for the volatility binary string. While remaining consistent with our results for two-stage coding, we have saved 7 bits on both sequences. So far, however, we have yet to design a coding scheme that costs less than $n = 6{,}429$ bits for the return indicators.

Although many mixture codes can be created by making different choices for the prior density assigned to p, the distribution $m(\cdot)$ is guaranteed to have a closed-form expression only for a family of so-called conjugate priors. In general, numerical or Monte Carlo methods might be necessary to evaluate the code length of a mixture code.

Predictive Coding. Imagine that the up-and-down information for the return series was to be sent to San Diego on a daily basis, and assume that the sender and receiver have agreed to use a fixed code on $\{0, 1\}$. For simplicity, suppose that they have decided on a Bernoulli encoder with $p = 1/2$. Each day, a new indicator is generated and sent to San Diego at a cost of $-\log_2(1/2) = 1$ bit. For the following 6,429 days, this would total 6,429 bits. (This is equivalent to simply listing the data without introducing a model.) Such a coding scheme could not be very economical if, on average, the number of "up days" was much smaller than the number of "down days" or vice versa. If instead we postulate an iid Bernoulli model with an unknown probability p, then all of the previous information, known to both sender and receiver, can be used to possibly improve the code length needed to transmit the sequence. Suppose that over the past $t-1$ days, k_{t-1} ups or 1's have been accumulated. At day t, a new Bernoulli coder can be used with the Laplace estimator $\hat{p}_{t-1} = (k_{t-1}+1)/(t+1)$, avoiding difficulties when $k_{t-1} = 0$ or $t-1$. At the outset, sender and receiver agree to take $p_0 = 1/2$. If on day t we see an increase in the return of the DJIA, then the Bernoulli coder with $p = \hat{p}_{t-1}$ is used at a cost of $L_t(1) = -\log_2 \hat{p}_{t-1}$ bits. Otherwise, we transmit a 0, requiring $L_t(0) = -\log_2(1-\hat{p}_{t-1})$ bits.

This accounting makes use of so-called "fractional bits." In practical terms, it is not possible to send less than a single bit of information per day. But if we delay transmission by several days, then we can send a larger piece of the data at a much lower cost. When the delay is n days, this "predictive" method is equivalent to the batch scheme used in mixture coding (sending the entire data string at once). We have chosen to sidestep this important practical complication and instead present predictive coding as if it could be implemented on a daily basis. The broad concept is important here, as it is similar to other frameworks for statistical estimation, including P. Dawid's prequential analysis.

For a string $x^n = (x_1, \ldots, x_n)$ with k 1's and $(n-k)$ 0's, the total code length over 6,429 days is

$$\sum_{t=1}^{n} L_t(x_t).$$

Equivalently, a joint probability distribution on $\{0, 1\}^n$ has been constructed predictively:

$$q(x^n) = \prod_{t=1}^{n} \hat{p}_{t-1}^{x_t}(1-\hat{p}_{t-1})^{1-x_t}, \quad (9)$$

where

$$-\log_2 q(x^n) = \sum_{t=1}^{n} L_t(x_t).$$

Rewriting (9), we find that

$$\begin{aligned} -\log_2 q(x^n) &= -\sum_{t=1}^{n}\left[x_t \log_2 \hat{p}_{t-1} + (1-x_t)\log_2(1-\hat{p}_{t-1})\right] \\ &= -\sum_{t:x_t=1} \log_2 \hat{p}_{t-1} - \sum_{t:x_t=0} \log_2(1-\hat{p}_{t-1}) \\ &= -\sum_{t:x_t=1} \log_2(k_{t-1}+1) - \sum_{t:x_t=0} \log_2(t-k_{t-1}) \\ &\quad + \sum_{t=1}^{n} \log_2(t+1) \\ &= -\log_2 k! - \log_2(n-k)! + \log_2(n+1)!, \end{aligned}$$

which is exactly the same expression as (8), the code length derived for the uniform mixture code (an unexpected equivalence to which we return shortly). Although the bits are counted differently, the code lengths are the same. Thus, from the previous example, the predictive code lengths are 6,434 bits and 5,782 bits for the return and volatility strings. In some sense, the predictive coder is designed to learn about p from the past up-and-down information and hence improves the encoding of the next day's indicator. This form of coding has intimate connections with machine learning with its focus on accumulative prediction error (see Haussler, Kearns, and Schapire 1994) and the prequential approach of Dawid (1984, 1991). Clearly, predictive coding requires an ordering of the data that is very natural in on-line transmission and time series models, but conceptually less appealing in other contexts like multivariate regression. As in this case, however, when a proper Bayes estimator is used in the predictive coder, the ordering can sometimes disappear in the final expression for code length. A proof of this somewhat surprising equivalence between predictive and mixture code lengths has been given by Yu and Speed (1992) for a general multinomial model.

In the time series context, predictive coding offers us the ability to easily adapt to nonstationarity in the data source, a tremendous advantage over the other schemes discussed so far. For example, suppose that we use only the number of ups encountered in the last 1,000 days to estimate p in a Bernoulli model for the next day's indicator. When applied to the volatility difference indicator series, we save only 3 bits over the 5,782 needed for the simple predictive coder, implying that this string is fairly stationary. To explore the possible dependence structure in the volatility difference indicator string, we postulated a first-order Markov model, estimating the transition probabilities from the indicators for the last 1,000 days. Under this scheme, we incur a cost of 5,774 bits.

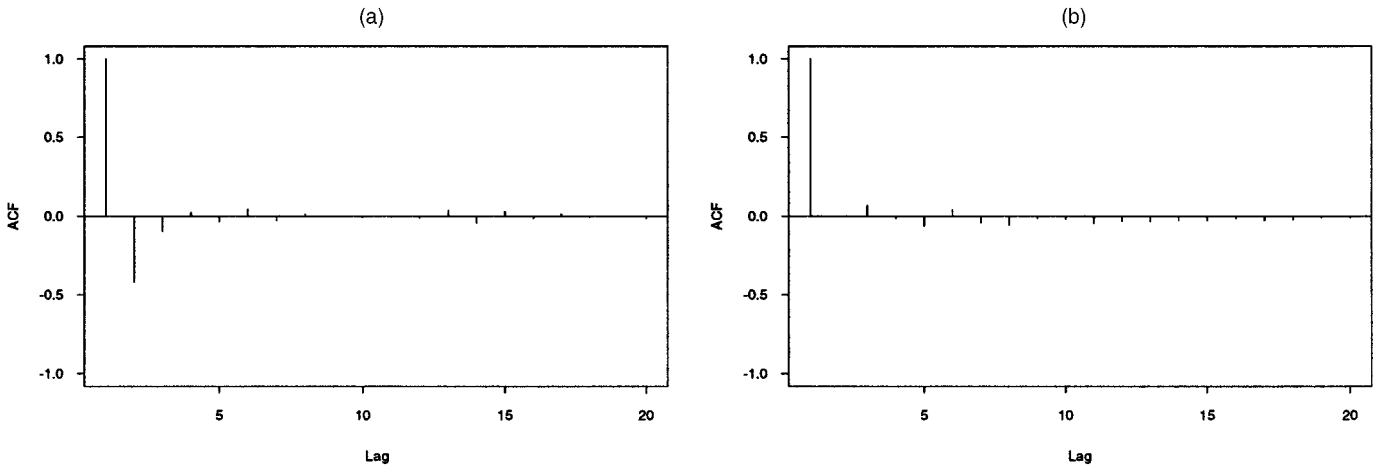

Figure 3. Autocorrelation Functions for the Differences of the Return (a) and Volatility (b) Series. With 6,429 points, the usual confidence intervals barely appear as distinct from the solid line $y = 0$.

Such a small decrease is evidence that there is little dependence in this string, and that the biggest saving in terms of code length comes from learning the underlying probability p in an iid Bernoulli model. This is because the volatility difference series $V_t - V_{t-1}$ exhibits very little correlation structure, despite the fact that volatility series itself is an exponentially weighted moving average. Figure 3 plots the autorcorrelation function for each of the differenced volatility and return series. In terms of the derived up-and-down indicators, the volatility string has a first-order autocorrelation of $-.02$, practically nonexistent.

The indicator string derived from the return series is a different story, however. As with the volatility string, estimating p based on the previous 1,000 days of data does not result in a smaller code length, suggesting little nonstationarity. However, there is considerably more dependence in the return string. Although the underlying series R_t has little autocorrelation structure, the differences $R_t - R_{t-1}$ exhibit a large dependence at a lag of 1 (see Fig. 3). The first-order autocorrelation in the return difference indicator string is $-.42$, indicating that our Markov model might be more effective here than for the volatility string. In fact, by postulating a first-order Markov model (estimating transition probabilities at time t from all of the previous data), we reduce the code length to 6,181, a 4% or 253-bit saving over the 6,434 bits required for the simple predictive coder. By instead estimating the transition probabilities from the last 1,000 days of data, we can produce a further decrease of only 10 bits, confirming our belief that the return difference indicator string is fairly stationary. Under this coding strategy, we are finally able to transmit the return string using fewer than $n = 6{,}429$ bits. In general, predictive coding can save in terms of code length even when an iid model is considered. When dependence or nonstationarity are present, we can experience even greater gains by directly modeling such effects, say through a Markov model. Of course, with some effort the two-stage and mixture coding schemes can also incorporate these features, and we should see similar code length reductions when the data support the added structure.

2.3 The Miminum Description Length Principle

In the previous two sections we motivated the code length interpretation of probability distributions and illustrated the use of models for building good codes. Although our focus was on compression, motivation for the MDL principle can be found throughout Example 4; probability models for each binary string were evaluated on the basis of their code length. In statistical applications, postulated models help us make inferences about data. The MDL principle in this context suggests choosing the model that provides the shortest description of our data. For the purpose of this article, the act of describing data is formally equivalent to coding. Thus, when applying MDL, our focus is on casting statistical modeling as a means of generating codes, and the resulting code lengths provide a metric by which we can compare competing models. As we found in Example 4, we can compute a code length without actually exhibiting a code (i.e., generating the map between data values and code words), making the implementation details somewhat unimportant.

As a broad principle, MDL has rich connections with more traditional frameworks for statistical estimation. In classical parametric statistics, for example, we want to estimate the parameter θ of a given model (class)

$$\mathcal{M} = \left\{ f(x^n|\theta) : \theta \in \Theta \subset \mathbb{R}^k \right\}$$

based on observations $x^n = (x_1, \ldots, x_n)$. The most popular estimation technique in this context is derived from the Maximum Likelihood Principle (ML principle) pioneered by R. A. Fisher (cf., Edwards 1972). Estimates $\hat{\theta}_n$ are chosen so as to maximize $f_\theta(x^n)$ over $\theta \in \Theta$. As a principle, ML is backed by $\hat{\theta}_n$'s asymptotic efficiency in the repeated-sampling paradigm (under some regularity conditions) and its attainment of the Cramer–Rao information lower bound in many exponential family examples (in the finite-sample case). From a coding perspective, assume that both sender and receiver know which member f_θ of the parametric family $\mathcal{M}$ generated a data string x^n (or, equivalently, both sides know θ). Then Shannon's Source Coding Theorem states that the best description length of x^n (in an average sense) is simply $-\log f_\theta(x^n)$, because on

average the code based on f_θ achieves the entropy lower bound (5). Obviously, minimizing $-\log_2 f_\theta(x^n)$ is the same as maximizing $f_\theta(x^n)$, so that MDL coincides with ML in parametric estimation problems. Thus in this setting, MDL enjoys all of the desirable properties of ML mentioned earlier. In modeling applications like those discussed in Example 4, however, we had to transmit θ, because the receiver did not know its value in advance. Adding in this cost, we arrive at a code length

$$-\log f_\theta(x^n) + L(\theta)$$

for the data string x^n. Now if each parameter value requires the same fixed number of bits to transmit, or rather if $L(\theta)$ is constant, then the MDL principle seeks a model that minimizes $-\log f_\theta(x^n)$ among all densities in the family. (This is the case if we transmit each value of θ with a fixed precision.)

It is well known, however, that ML breaks down when one is forced to choose among nested classes of parametric models. This occurs most noticeably in variable selection for linear regression. The simplest and most illustrative selection problem of this type can be cast as an exercise in hypothesis testing.

Example 5. Assume that $x^n = (x_1, \ldots, x_n)$ are n iid observations $N(\theta, 1)$ for some $\theta \in \mathbb{R}^1$, and that we want to test the hypothesis $H_0 : \theta = 0$ versus $H_1 : \theta \neq 0$. Equivalently, we want to choose between the models

$$\mathcal{M}_0 = \{N(0, 1)\} \quad \text{and} \quad \mathcal{M}_1 = \{N(\theta, 1) : \theta \neq 0\}$$

on the basis of x^n. In this case, if we maximize the likelihoods of both models and choose the one with the larger maximized likelihood, then $\mathcal{M}_1$ is always chosen unless $\bar{x}_n = 0$, an event with probability 0 even when $\mathcal{M}_0$ is true.

Note that ML has no problem with the estimation of θ if we merge the two model classes $\mathcal{M}_0$ and $\mathcal{M}_1$. Clearly the formulation of the model selection problem is responsible for the poor performance of ML. To be fair, the ML principle was developed only for a single parametric family, and hence it is not guaranteed to yield a sensible selection criterion.

The Bayesian approach to statistics has a natural solution to this selection problem. After assigning a prior probability distribution to each model class, the Bayesian appeals to the posterior probabilities of these classes to select a model (see, e.g., Bernardo and Smith 1994). Given the formulation of the foregoing problem, the assignment of priors is a subjective matter, which in recent years has been made increasingly on the basis of computational efficiency. Some attempts have been made to reduce the level of subjectivity required for such an analysis, producing "automatic" or "quasi-automatic" Bayesian procedures (Berger and Pericchi 1996; O'Hagan 1995). A simple solution involves use of BIC, an approximation to the posterior distribution on model classes derived by Schwarz (1978). Although based on the assumption that proper priors have been assigned to each class, this approximation effectively eliminates any explicit dependence on prior choice. The resulting selection rule takes on the form of a penalized log-likelihood, $-\log f_{\hat{\theta}_n}(x^n) + k/2 \log n$, where $\hat{\theta}_n$ is the ML estimate of the k-dimensional parameter θ.

To repair ML in this context, recall that Fisher first derived the likelihood principle within a single parametric family, starting from a Bayesian framework and placing a uniform prior on the parameter space (Edwards 1972). Let $L_{\mathcal{M}}$ denote the description length of a data string x^n based on a single family or model (class) $\mathcal{M}$. Because MDL coincides with ML when choosing among members of $\mathcal{M}$, we can think of $2^{-L_{\mathcal{M}}}$ as the "likelihood" of the class given x^n. Now, applying Fisher's line of reasoning to models, we assign a uniform prior on different families and maximize the newly defined "likelihood." This yields the principle of MDL for model selection.

In Example 4, however, we presented several different coding schemes that can be used to define the description length $L_{\mathcal{M}}$ of a given model class $\mathcal{M}$. Although many more schemes are possible, not all of these are usable for statistical model selection. As our emphasis is on a coding *interpretation*, we want to know under what general conditions these schemes provide us with "valid" description lengths based on $\mathcal{M}$ (in the sense that they yield selection rules with provably good performance). At an intuitive level, we should select a code that adequately represents the knowledge contained in a given model class, a notion that we make precise in Section 5. Rissanen's (1986a) pointwise lower bound on the redundancy for parametric families is a landmark for characterizing the statistical properties of MDL criteria. Roughly, the expected redundancy of a code corresponds to the price that one must pay for not knowing which member of the model class generated the data x^n. Rissanen (1986a) demonstrated that for a regular parametric family of dimension k, this amounts to at least $k/2 \log n$ extra bits. Any code length that achieves this lower bound qualifies (to first order in the parametric case) as a valid description length of the model class given a data string x^n, and the associated model selection criteria have good theoretical properties.

An alternative measure for studying description length comes from a minimax lower bound on redundancy derived by Clarke and Barron (1990). Both the pointwise and minimax lower bounds not only make compelling the use of MDL in statistical model selection problems, but also extend Shannon's source coding theorem to so-called *universal coding*, where the source or true distribution is only known to belong to a parametric family. A more rigorous treatment of this theoretical material is presented in Section 5. It follows from these results that $-\log f_{\hat{\theta}_n}(x^n) + k/2 \log n$ (modular a constant term) is a valid code length for our parametric family introduced at the beginning of this section. We recognize this expression as BIC. More careful asymptotics yields a tighter bound on redundancy that can be met only if Jeffreys's prior is integrable in the particular family under study (see Barron et al. 1998).

The appearance of BIC as a valid code length and the more refined result about Jeffreys's prior are just two of a number of connections between MDL and Bayesian statistics. Among the various forms of MDL presented in Example 4, mixture coding bears the closest direct resemblance to a Bayesian analysis. For example, both frameworks can depend heavily on the assignment of priors, and both are subject to the requirement that the corresponding marginal (or predictive) distribution of a data string is integrable. When this integrabil-

ity condition is not met, the Bayesian is left with an indeterminant Bayes factor, and the connection with prefix coding is lost (as Kraft's inequality is violated). [This situation is most commonly encountered under the assignment of so-called weak prior information that leaves the marginal distribution improper. For example, as improper priors are specified only up to a multiplicative constant, the associated Bayes factor (a ratio of predictive or marginal densities) inherits an unspecified constant.] Both schemes also benefit from "realistic" priors, although the classes entertained in applications tend to be quite different. [MDL has found wide application in various branches of engineering. For the most part, Rissanen's reasoning is followed "in spirit" to derive effective selection criteria for the problem at hand. New and novel applications of MDL include generating codes for trees for wavelet denoising (Moulin, 1996; Saito, 1994).] In terms of loss functions, because MDL minimizes the mixture code length, it coincides with a Maximum a Posteriori (MAP) estimate derived using 0–1 loss. MDL parts company with Bayesian model selection in the treatment of hyperparameters that accompany a prior specification. Rissanen (1989) proposed a (penalized) ML approach that we examine in detail in Section 4.1.1 for ordinary regression problems. Also, given Kraft's inequality, MDL technically allows for subdistributions. In applications involving discrete data, the only available coding scheme often does not sum to 1 or, equivalently, is not Kraft-tight.

In addition to mixture MDL, we have applied both two-stage and predictive coding schemes to the indicator series from Example 4. In the next section we introduce one more code based on the so-called normalized ML. Although these forms do not have explicit Bayesian equivalents, they can be thought of as building a marginal density over a model class or parametric family that is independent of the parameters. Hence when the code for the model class corresponds to a proper distribution, or is Kraft-tight, one can borrow Bayesian tools to assess uncertainty among candidate models. (This type of analysis has not been explored in the MDL literature.) In general, MDL formally shares many aspects of both frequentist and Bayesian approaches to statistical estimation. As Rissanen has noted, MDL provides an objective and welcome platform from which to compare (possibly quite disparate) model selection criteria. We are confident that the rich connections between information theory and statistics will continue to produce new forms of MDL as the framework is applied to increasingly challenging problems.

3. DIFFERENT FORMS OF DESCRIPTION LENGTH BASED ON A MODEL

In this section we formally introduce several coding schemes that provide valid description lengths of a data string based on classes of probability models, in the sense that they achieve the universal coding lower bound to the $\log n$ order (cf. Sec. 5). We use the description lengths discussed here in our implementation of MDL for the model selection problems in Sections 4 and 5. Three of these schemes were introduced in Example 4 for compression purposes. In that case, probability models helped us build codes that could be used to communicate data strings with as few bits as possible. The only necessary motivation for enlisting candidate models was that they provided short descriptions of the data. In statistical applications, however, probability distributions are the basis for making inference about data, and hence play a more refined role in modeling. In this section we follow the frequentist philosophy that probability models (approximately) describe the mechanism by which the data are generated.

Throughout this section, we focus mainly on a simple parametric model class $\mathcal{M}$ comprising a family of distributions indexed by a parameter $\theta \in \mathbb{R}^k$. Keep in mind, however, that the strength of the MDL principle is that it can be successfully applied in far less restrictive settings. Let $x^n = (x_1, x_2, \ldots, x_n)$ denote a data string, and recall our model class

$$\mathcal{M} = \{f(x^n|\theta) : \theta \in \Theta \subset \mathbb{R}^k\}.$$

For convenience, we consider coding schemes for data transmission, so that when deriving code or description lengths for x^n based on $\mathcal{M}$, we can assume that $\mathcal{M}$ is known to both sender and receiver. If this were not the case, then we would also have to encode information about $\mathcal{M}$, adding to our description length. Finally, we calculate code lengths using the natural logarithm log, rather than $\log_2$ as we did in the previous section. The unit of length is now the *nat*.

Next we revisit the three coding schemes introduced briefly in Example 2.2. We derive each in considerably more generality and apply them to the hypothesis testing problem of Example 4. Building on this framework, in Section 4 we provide a rather extensive treatment of MDL for model selection in ordinary linear regression. A rigorous justification of these procedures is postponed to Section 5. There, we demonstrate that in the simple case of a parametric family, these coding schemes give rise to code lengths that all achieve (to first order) both Rissanen's pointwise lower bound on redundancy and the minimax lower bound covered in Section 5 (Clarke and Barron 1990). This implies that these schemes produce valid description lengths, each yielding a usable model selection criterion via the MDL principle.

3.1 Two-Stage Description Length

To a statistical audience, the two-stage coding scheme is perhaps the most natural method for devising a prefix code for a data string x^n. We first choose a member of the class $\mathcal{M}$, and then use this distribution to encode x^n. Because we are dealing with a parametric family, this selection is made via an estimator $\hat{\theta}_n$, after which a prefix code is built from $f_{\hat{\theta}_n}$. Ultimately, the code length associated with this scheme takes the form of a penalized likelihood, the penalty being the cost to encode the estimated parameter values $\hat{\theta}_n$.

Stage 1: The Description Length $L(\hat{\theta}_n)$ for the Estimated Member $\hat{\theta}_n$ of the Model Class. In the first stage of this coding scheme, we communicate an estimate $\hat{\theta}_n$ obtained by, say, ML or some Bayes procedure. This can be done by first discretizing a compact parameter space with precision $\delta_m = 1/\sqrt{n}$ (m for the model) for each member of θ, and then transmitting $\hat{\theta}_n$ with a uniform encoder. Rissanen (1983, 1989) showed that this choice of precision is optimal in regular parametric families. The intuitive argument is that $1/\sqrt{n}$ represents the magnitude of the estimation error in $\hat{\theta}_n$ and hence there is no need to encode the estimator with greater

precision. In general, our uniform encoder should reflect the convergence rate of the estimator we choose for this stage. Assuming the standard parametric rate $1/\sqrt{n}$, we pay a total of $-k \log 1/\sqrt{n} = k/2 \log n$ nats to communicate an estimated parameter θ_n of dimension k.

Although the uniform encoder is a convenient choice, we can take any continuous distribution w on the parameter space and build a code for $\hat{\theta}_n$ by again discretizing $\delta_m = 1/\sqrt{n}$ with the same precision:

$$L(\hat{\theta}_n) = -\log w([\hat{\theta}_n]_{\delta_m}) + \frac{k}{2}\log n,$$

where $[\hat{\theta}_n]_{\delta_m}$ is $\hat{\theta}_n$ truncated to precision δ_m. In the MDL paradigm, the distribution w is introduced as an ingredient in the coding scheme, not as a Bayesian prior. But if we have reason to believe that a particular prior w reflects the likely distribution of the parameter values, then choosing w for description purposes is certainly consistent with Shannon's Source Coding Theorem. Clearly, both recipes lead to description lengths with the same first-order term,

$$L(\hat{\theta}_n) \approx \frac{k}{2}\log n,$$

where k is the Euclidean dimension of the parameter space.

Stage 2: The Description Length of Data Based on the Transmitted Distribution. In the second stage of this scheme, we encode the actual data string $x^n = (x_1, \ldots, x_n)$, using the distribution indexed by $[\hat{\theta}_n]_{\delta_m}$. For continuous data, we follow the prescription in Section 2.1.2, discretizing the selected distribution with precision δ_d (d for the data). In this stage we can take δ_d to be machine precision. The description length for coding x^n is then

$$-\log f(x_1, \ldots, x_n | [\hat{\theta}_n]_{\delta_m}) - n \log \delta_d.$$

When the likelihood surface is smooth as in regular parametric families, the difference

$$\log f(x_1, \ldots, x_n | [\hat{\theta}_n]_{\delta_m}) - \log f(x_1, \ldots, x_n | \hat{\theta}_n)$$

is of a smaller order of magnitude than the model description length $k/2 \log n$. In addition, the quantity $n \log \delta_d$ is constant for all the models in $\mathcal{M}$. Hence we often take

$$-\log f(x_1, \ldots, x_n | \hat{\theta}_n),$$

the negative of the maximized log-likelihood for the maximum likelihood estimator (MLE) $\hat{\theta}_n$, as the simplified description length for a data string x^n based on $f(\cdot|\hat{\theta}_n)$.

Combining the code or description lengths from the two stages of this coding scheme, we find that for regular parametric families of dimension k, the (simplified) two-stage MDL criterion takes the form of BIC,

$$-\log f(x_1, \ldots, x_n | \hat{\theta}_n) + \frac{k}{2}\log n. \tag{10}$$

Again, the first term represents the number of nats needed to encode the date sequence x^n given an estimate $\hat{\theta}_n$, whereas the second term represents the number of nats required to encode the k components of $\hat{\theta}_n$ to precision $1/\sqrt{n}$. It is worth noting that the simplified two-stage description length is valid if one starts with a $1/\sqrt{n}$-consistent estimator other than the MLE, even though traditionally only the MLE has been used. This is because only the rate of a $1/\sqrt{n}$ estimator is reflected in the $\log n$ term. In more complicated situations, such as the clustering analysis presented in Section 4, more than two stages of coding might be required.

Example 4 (Continued). Because $\mathcal{M}_0 = \{N(0, 1)\}$ consists of a single distribution, we know from Shannon's source coding theorem that the cost for encoding $x^n = (x_1, \ldots, x_n)$ is

$$L_0(x^n) = \frac{1}{2}\sum_{t=1}^{n} x_t^2 + \frac{n}{2}\log(2\pi).$$

Next, consider encoding x^n via a two-stage scheme based on the class

$$\mathcal{M}_1 = \{N(\theta, 1) : \theta \neq 0\}.$$

If we estimate θ by the MLE $\hat{\theta}_n = \bar{x}_n$, then the two-stage description length (10) takes the form

$$L_1(x^n) = \frac{1}{2}\sum_{t=1}^{n}(x_t - \bar{x}_n)^2 + \frac{n}{2}\log(2\pi) + \frac{1}{2}\log n. \tag{11}$$

Thus, following the MDL principle, we choose $\mathcal{M}_0$ over $\mathcal{M}_1$ based on the data string x^n if

$$|\bar{x}_n| < \sqrt{\log(n)/n}.$$

In this case the MDL criterion takes the form of a likelihood ratio test whose significance level shrinks to 0 as n tends to infinity.

3.2 Mixture Minimum Description Length and Stochastic Information Complexity

The mixture form of description length naturally lends itself to theoretical studies of MDL. In Section 5 we highlight connections between this form and both minimax theory and the notion of channel capacity in communication theory (Cover and Thomas 1991). Because mixture MDL involves integrating over model classes, it can be hard to implement in practice. To get around such difficulties, it can be shown that a first-order approximation to this form coincides with the two-stage MDL criterion derived earlier. The proof of this fact (Clarke and Barron 1990) mimics the original derivation of BIC as an approximate Bayesian model selection criterion (Schwarz 1978; Kass and Raftery 1995). An alternative approximation yields yet another form of description length known as Stochastic Information Complexity (SIC). As we demonstrate, mixture MDL shares many formal elements with Bayesian model selection because the underlying analytical tools are the same. However, the philosophies behind each approach are much different. In the next section we explore how these differences translate into methodology in the context of ordinary linear regression.

The name "mixture" for this form reveals it all. We base our description of a data string x^n on a distribution obtained

by taking a mixture of the members in the family with respect to a probability density function w on the parameters,

$$m(x^n) = \int f_\theta(x^n) w(\theta)\, d\theta. \tag{12}$$

Again, we introduce w not as a prior in the Bayesian sense, but rather as a device for creating a distribution for the data based on the model class $\mathcal{M}$. Given a precision δ_d, we follow Section 2.1.2 and obtain the description length

$$-\log m(x^n) = -\log \int f(x_1, \ldots, x_n|\theta) w(\theta)\, d\theta + n \log \delta_d.$$

Ignoring the constant term, we arrive at

$$-\log \int f(x_1, \ldots, x_n|\theta) w(\theta)\, d\theta. \tag{13}$$

This integral has a closed-form expression when $f(\cdot|\theta)$ is an exponential family and w is a conjugate prior, as is the case in Example 4. When choosing between two models, the mixture form of MDL is equivalent to a Bayes factor (Kass and Raftery 1995) based on the same priors. A popular method for calculating Bayes factors involves using Markov chain Monte Carlo (McMC) (George and McCulloch 1997), which can be applied to obtain the description length of mixture codes.

Example 4 (Continued). If we put a Gaussian prior $w = N(0, \tau)$ on the mean parameter θ in $\mathcal{M}_1$ (note that τ is the variance), then we find that

$$-\log m(x^n) = \frac{n}{2}\log(2\pi) + \frac{1}{2}\log\det(I_n + \tau J_n) + \frac{1}{2} x_n'(I_n + \tau J_n)^{-1} x_n, \tag{14}$$

where I_n is the $n \times n$ identity matrix and J_n is the $n \times n$ matrix of 1's. Simplifying the foregoing expression, we arrive at

$$\frac{1}{2}\sum_t x_t^2 - \frac{1}{2}\frac{n}{1 + 1/(n\tau)}\bar{x}_n^2 + \frac{n}{2}\log(2\pi) + \frac{1}{2}\log(1 + n\tau). \tag{15}$$

Comparing this to the description length for the two-stage encoder (11), we find a difference in the penalty,

$$\frac{1}{2}\log(1 + n\tau), \tag{16}$$

which (to first order) is asymptotically the same as that associated with BIC, $1/2 \log n$. Depending on the value of the prior variance τ, (16) represents either a heavier ($\tau > 1$) or a lighter ($\tau < 1$) penalty. Figure 4 presents a graphical comparison for two values of τ.

An analytical approximation to the mixture $m(\cdot)$ in (12) is obtained by Laplace's expansion when w is smooth (Rissanen 1989). Essentially, we arrive at a two-stage description length, which we will call SIC:

$$\mathrm{SIC}(x^n) = -\log f(x^n|\hat{\theta}_n) + \frac{1}{2}\log\det(\hat{\Sigma}_n), \tag{17}$$

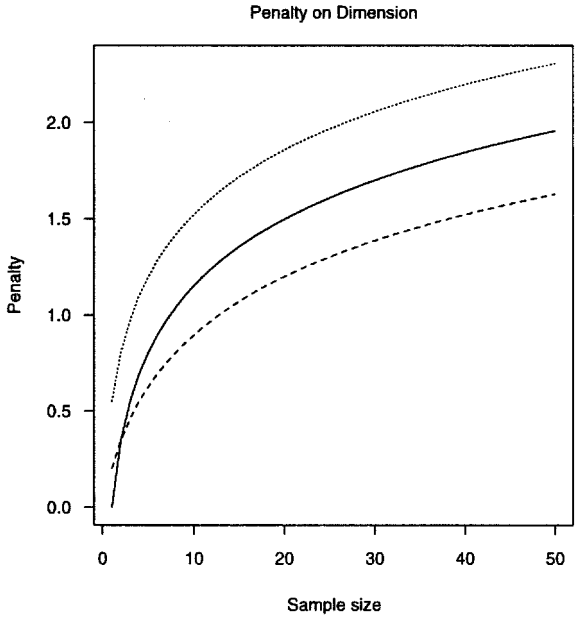

Figure 4. Comparison of the Penalties Imposed by BIC (——) *and the Mixture Form of MDL for* $\tau = .5$ (– – –) *and* $\tau = 2$ (· · · ·). *The sample size* n *ranges from 1 to 50.*

where $\hat{\theta}_n$ is the MLE and $\hat{\Sigma}_n$ is the Hessian matrix of $-\log f(x^n|\theta)$ evaluated at $\hat{\theta}_n$. For iid observations from a regular parametric family, and as $n \to \infty$,

$$\frac{1}{2}\log\det(\hat{\Sigma}_n) = \frac{1}{2}\log\det(nI(\hat{\theta}_n))(1 + o(1)) = \frac{k}{2}\log n(1 + o(1)). \tag{18}$$

Here $I(\cdot)$ is the Fisher information matrix of a single observation. The middle term in this chain of equalities,

$$\frac{1}{2}\log\det(nI(\hat{\theta})), \tag{19}$$

can be interpreted as the number of nats needed to encode the k estimated parameter values if we discretize the jth parameter component with a precision $\mathrm{SE}(\hat{\theta}_j) = 1/\sqrt{nI_{jj}(\theta)}$ (provided that the estimated parameters are either independent or the discretization is done after the parameter space is transformed so that the information matrix under the new parameterization is diagonal). It is obviously sensible to take into account the full estimation error, and not just the rate, when discretizing. The final equality in (18) tells us that in the limit, SIC is approximately BIC or two-stage MDL. For finite sample sizes, however, SIC's penalty term is usually not as severe as BIC's, and hence in some situations, SIC outperforms BIC. Rissanen (1989, p. 151, table 6) illustrated this difference by demonstrating that SIC outperforms two-stage MDL when selecting the order in an autoregression model with $n = 50$. In Section 4 we present many more such comparisons in the context of ordinary linear regression.

3.3 Predictive Description Length

Any joint distribution $q(\cdot)$ of $x^n = (x_1, \ldots, x_n)$ can be written in its *predictive form*,

$$q(x^n) = \prod_{t=1}^{n} q(x_t | x_1, \ldots, x_{t-1}).$$

Conversely, given a model class $\mathcal{M}$, it is a simple matter to obtain a joint distribution for x^n given a series of predictive distributions. In many statistical models, the conditionals $f_\theta(x_j | x_1, \ldots, x_{j-1})$ share the same parameter θ. [Typically, $f(x_1) = f_0(x_1)$ will not depend on θ, however.] For iid data generated from a parametric family $\mathcal{M}$, this is clearly the case. Other applications where this property holds include time series, regression, and generalized linear models. Suppose that for each t we form an estimate $\hat{\theta}_{t-1}$ from the first $(t-1)$ elements of x^n. Then the expression

$$q(x_1, \ldots, x_n) = \prod_t f_{\hat{\theta}_{t-1}}(x_t | x_1, \ldots, x_{t-1}) \tag{20}$$

represents a joint distribution based on the model class $\mathcal{M}$ that is free of unknown parameters. The cost of encoding a data string x^n using (20) is

$$-\log q(x_1, \ldots, x_n) = -\sum_t \log f_{\hat{\theta}_{t-1}}(x_t | x_1, \ldots, x_{t-1}). \tag{21}$$

The MDL model selection criterion based on this form of description is called PMDL for its use of the predictive distribution (20). PMDL is especially useful for time series models (Hannan and Rissanen 1982; Hannan et al. 1989; Huang 1990).

By design, predictive MDL is well suited for time series analysis, where there is a natural ordering of the data; on-line estimation problems in signal processing; and on-line data transmission applications like the binary string example discussed Section 2. At a practical level, under this framework both sender and receiver start with a predetermined encoder f_0 to transmit the first data point x_1. This accounts for the leading term in the summation (21). At time t, because the previous $(t-1)$ points are known at each end of the channel, the distribution $f_{\hat{\theta}_{t-1}}(x_t | x_1, \ldots, x_{t-1})$ is also known. This is the tth term in the summation (21). By using the predictive distributions to sequentially update the code, both the encoder and decoder are in effect learning about the true parameter value and hence can do a better job of coding the data string (provided that one member of the model class actually generated the data).

Example 4 (Continued). If we take the initial density f_0 as $N(0, 1)$ and set

$$\hat{\theta}_{t-1} = \bar{x}_{t-1} = \frac{1}{t-1} \sum_{i=1}^{t-1} x_i$$

(with $\bar{x}_0 = 0$) based on $\mathcal{M}_1$, then

$$\begin{aligned} -\log q(x^n) &= -\sum_{t=1}^{n} \log f_{\hat{\theta}_{t-1}}(x_t | x^{t-1}) \\ &= \frac{n}{2} \log(2\pi) + \frac{1}{2} \sum_{t=1}^{n} (x_t - \bar{x}_{t-1})^2. \end{aligned} \tag{22}$$

The reasoning that we followed in deriving PMDL is identical to the prequential approach to statistics advocated by Dawid (1984, 1991). The form (21) appeared in the literature on Gaussian regression and time series analysis as the *predictive least squares criterion* long before the development of MDL, and early work on PMDL focused mainly on these two applications. The interested reader is referred to Hannan and Rissanen 1982; Hannan and Kavalieris 1984; Rissanen 1986b; Hannan et al. 1989; Hemerly and Davis 1989; Wei 1992; Gerencsér 1994; Speed and Yu 1994. The recent results of Qian, Gabor, and Gupta (1996) extended the horizon of this form of MDL to generalized linear models.

In Section 4 we illustrate the application of PMDL to the (differenced) daily return series studied in Example 3. In this case we work with the "raw" data rather than the binary up-and-down string treated earlier. Although in special cases, such as multinomial, the ordering disappears when a Bayes estimator is used for the prediction, in general PMDL depends on a sensible ordering of the data. It is not clear how useful it will be in, say, multivariate regression problems. To get around this problem, Rissanen (1986b) suggested repeatedly permuting the data before applying PMDL, and then averaging the predictive code lengths. In Section 4 we avoid these complications and discuss PMDL only in the context of time series data.

3.4 Other Forms of Description Length

The MDL principle offers the opportunity to develop many other forms of description length besides the three discussed earlier. In Section 5 we present some of the theoretical validation required for new coding schemes or, equivalently, new MDL criteria. For example, weighted averages or mixtures of the three common forms will give rise to new description lengths that all achieve the pointwise and minimax lower bounds on redundancy and hence can be used for model selection. Further investigation is required to determine how to choose these weights in different modeling contexts.

Recently, Rissanen (1996) developed an MDL criterion based on the normalized maximum likelihood (NML) coding scheme of Shtarkov (1987) (Barron et al. 1998). For a flavor of how it was derived, we apply NML to the binary DJIA up-and-down indicators introduced in Section 2.

Example 3 (Continued): Normalized Maximum Likelihood Coding. As was done in the two-stage scheme, we first transmit k. Then both sender and receiver know that the indicator sequence must be among the collection of strings of size n with exactly k 1's. This group of sequences is known as the *type class* $T(n, k)$. Under the iid Bernoulli model, each string in the type class is equally likely, and we can use a uniform code on $T(n, k)$ to communicate its elements. When applied to the return string, the NML code requires $\log_2 \frac{n!}{k!(n-k)!}$, or 6,421 bits, giving us a total code length of 6,434 bits when we add the cost of encoding k. This represents a saving of 7 bits over the two-stage encoder described in Section 2, in which x^n was transmitted using an iid Bernoulli encoder with $\hat{p}_n = k/n$ in the second stage.

In general, the NML description of a data string works by restricting the second stage of coding to a data region identified by the parameter estimate. In the foregoing example, this

meant coding the return string as an element of $T(n, k)$ rather than $\{0, 1\}^n$. Rissanen (1996) formally introduced this scheme for MDL model selection and discussed its connection with minimax theory. We explore another application of this code when we take up ordinary linear regression in the next section.

4. APPLICATIONS OF MINIMUM DESCRIPTION LENGTH IN MODEL SELECTION

4.1 Linear Regression Models

Regression analysis is a tool for investigating the dependence of a random variable y on a collection of potential predictors $x_1, \ldots, x_M$. Associate with each predictor x_m a binary variable γ_m, and consider models given by

$$y = \sum_{\gamma_m=1} \beta_m x_m + \epsilon, \tag{23}$$

where ϵ has a Gaussian distribution with mean 0 and unknown variance σ^2. The vector $\gamma = (\gamma_1, \ldots, \gamma_M) \in \{0, 1\}^M$ is used as a simple index for the 2^M possible models given by (23). Let β_γ and X_γ denote the vector of coefficients and the design matrix associated with those variables x_m for which $\gamma_m = 1$. In this section we apply MDL to the problem of model selection or, equivalently, the problem of identifying one or more vectors γ that yield the "best" or "nearly best" models for y in (23). In many cases, not all of the 2^M possibilities make sense, and hence our search might be confined to only a subset of index vectors γ.

The concept of "best," or more precisely, the measure by which we compare the performance of different selection criteria, is open to debate. Theoretical studies, for example, have examined procedures in terms of either consistency (in the sense that we select a "true" model with high probability) or prediction accuracy (providing small mean squared error), and different criteria can be recommended depending on the chosen framework. Ultimately, no matter how we settle the notion of "best," the benefit of a selection rule is derived from the insights that it provides into real problems. Mallows (1973) put it succinctly: "The greatest value of the device [model selection] is that it helps the statistician to examine some aspects of the structure of his data and helps him to recognize the ambiguities that confront him." In general, we should apply any selection procedure with some care, examining the structure of several good-fitting models rather than restricting our attention to a single "best" model. This point tends to be lost in simulation studies that necessitate blunt optimization of the criterion being examined.

At the end of this section, we present two applications that illustrate different practical aspects of model selection for regression analysis. The first application involves the identification of genetic loci associated with the inheritance of a given trait in fruit flies. Here MDL aids in evaluating specific scientific hypotheses. In the second application, we construct efficient representations for a large collection of hyperspectral (curve) data collected from common supermarket produce. Model selection is used in this context as a tool for data (dimension) reduction before application of (MDL-like) cluster analysis.

Our review of regression problems draws from various sources on MDL (Barron et al. 1998; Rissanen 1987, 1989; Speed and Yu 1993) as well as from the literature on Bayesian variable selection (George and McCulloch 1997; Kass and Raftery 1995; O'Hagan 1994; Smith and Spiegelhalter 1980). Because the need for selection in this context arises frequently in applications, we derive several MDL criteria in detail.

4.1.1 Several Forms of Minimum Description Length for Regression. Following the general recipe given in the previous sections, the MDL criteria that we derive for regression can all be written as a sum of two code lengths,

$$L(y|X_\gamma, \gamma) + L(\gamma). \tag{24}$$

This two-stage approach (see Sec. 3.1) explicitly combines both the cost to encode the observed data y using a given model γ and the cost to transmit our choice of model. For the second term, we use the Bernoulli(1/2) model discussed in Section 2.2 to describe the elements of γ; that is, the γ_m are taken be independent binary random variables and the probability that $\gamma_m = 1$ is 50%. Following this approach, each value of γ has the same probability,

$$P(\gamma) = \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right)^{M-k} = \left(\frac{1}{2}\right)^M. \tag{25}$$

Thus the cost $L(\gamma) = -\log P(\gamma)$ is constant. When we have reason to believe that smaller or larger models are preferable, a different Bernoulli model (with a smaller or larger value of p) can be used to encode γ. This approach has been taken in the context of Bayesian model selection and is discussed at the end of this section.

Having settled on this component in the code length, we turn our attention to the first term in (24), the cost of encoding the data, $L(y|X_\gamma, \gamma)$. We next will describe different MDL schemes for computing this quantity. To simplify notation, we drop the dependence on model index. Pick a vector γ and let $\beta = \beta_\gamma$ denote the $k = k_\gamma$ coefficients in (23) for which $\gamma_m = 1$. Similarly, let $X = X_\gamma$ be the design matrix associated with the selected variables in γ. For the most part, we work with ML estimates for both the regression coefficients β [also known as ordinary least squares (OLS) estimates] and the noise variance σ^2,

$$\hat{\beta} = (X'X)^{-1}X'y \qquad \text{and} \qquad \hat{\sigma}^2 = \|y - X\hat{\beta}\|^2/n. \tag{26}$$

Finally, we use RSS to represent the residual sum of squares corresponding to this choice of $\hat{\beta}$.

Two-Stage Minimum Description Length. Recall from Section 3.1 that two-stage MDL for a parametric model class is equivalent to BIC. Using the linear regression model (23), the code length associated with the observed data y is then given by the familiar forms

$$\frac{1}{2\sigma^2}\text{RSS} + \frac{k}{2}\log n \tag{27}$$

when σ^2 is known and

$$\frac{n}{2}\log\text{RSS} + \frac{k}{2}\log n \tag{28}$$

when it is unknown. To derive these expressions, we have applied the formula (10) using the estimators (26) and dropping constants that do not depend on our choice of model.

In both cases, the penalty applied to the dimension k depends on the sample size n. Related criteria like Mallows's C_p (Mallows 1973) and Akaike's AIC differ only in the size of this penalty:

$$C_p = \frac{1}{2\sigma^2}\text{RSS} + k \qquad \text{and} \qquad \text{AIC} = \frac{n}{2}\log \text{RSS} + k, \quad (29)$$

where we have again ignored terms that do not depend on our choice of model. [The form of C_p given here applies when σ^2 is known. If it is not known, Mallows (1973) suggested using an unbiased estimate $\widehat{\sigma}^2$.] While keeping the general form of these criteria, various authors have suggested other multipliers in front of k that can offer improved performance in special cases. (See Hurvich and Tsai 1989 and Sugiura 1978 for a corrected version of AIC for small samples; Hurvich, Simonoff, and Tsai 1998 for AIC in nonparametric regression; and Mallows 1995 for an interpretation of C_p when a different value of the penalty on model size is desired.) In Section 4.1.3 present an application in which a multiple of BIC penalty is proposed as the "correct" cost for a particular class of problems arising in genetics.

Mixture Minimum Description Length and Stochastic Information Complexity. In Section 3.2 we formally introduced the use of mixture distributions for constructing valid description lengths based on parametric classes. Because this form of MDL is structurally similar to a Bayesian analysis, our discussion of mixture MDL for regression problems is relatively brief and borrows heavily from a classical treatment of Bayesian variable selection for linear models. The framework for applying mixture codes in this context was given by Rissanen (1989).

Under the regression setup, we form a mixture distribution for y (conditional on our choice of model and the values of the predictors X) by introducing a density function $w(\beta, \sigma^2)$,

$$m(y|X) = \int f(y|X, \beta, \tau)\, w(\beta, \tau)\, d\beta\, d\tau. \quad (30)$$

To obtain a closed-form expression for $m(y|X)$, Rissanen (1989) took w as a member of the natural conjugate family of priors for the normal linear regression model (23), namely the so-called "normal inverse-gamma" distributions (see the Appendix). Under this density, the noise variance σ^2 is assigned an inverse-gamma distribution with shape parameter a. Then, conditional on σ^2, the coefficients β have a normal distribution with mean 0 and variance–covariance matrix $\sigma^2/c\Sigma$, where Σ is a known positive definite matrix. In his original derivation, Rissanen (1989) selected Σ to be the $k \times k$ identity matrix. Sidestepping this decision for the moment, the mixture code length for y computed from (13) is given by

$$\begin{aligned} -\log m(y|X) &= -\log m(y|X, a, c) \\ &= -\frac{1}{2}\log|c\Sigma^{-1}| + \frac{1}{2}\log|c\Sigma^{-1} + X^tX| \\ &\quad -\frac{1}{2}\log a + \frac{n+1}{2}\log(a + R_c), \end{aligned} \quad (31)$$

where

$$R_c = R_c = y^ty - y^tX\,(c\Sigma^{-1} + X^tX)^{-1}\,X^ty.$$

In (31) we have made explicit the dependence of the mixture code length on the values of two hyperparameters in the density w: a, the shape parameter of the inverse-gamma distribution for σ^2, and c, the (inverse) scale factor for β.

Rissanen (1989) addressed the issue of hyperparameters by choosing a and c to minimize the quantity (31) model by model. It is not difficult to see that $\hat{a} = R_c/n$, whereas for most values of Σ, $\hat{c}$ must be found numerically. An algorithm for doing this is given in the Appendix. By treating a and c in this way, however, we lose the interpretation of $-\log m(y|X, \hat{a}, \hat{c})$ as a description length. To remain faithful to the coding framework, the optimized hyperparameter values $\hat{a}$ and $\hat{c}$ must also be transmitted as overhead. Explicitly accounting for these extra factors yields the mixture code length

$$-\log m(y|X, \hat{a}, \hat{c}) + L(\hat{a}) + L(\hat{c}). \quad (32)$$

Because $\hat{a}$ and $\hat{c}$ are determined by maximizing the (mixture or marginal) log-likelihood (31), they can be seen to estimate a and c at the standard parametric rate of $1/\sqrt{n}$. Therefore, we take a two-stage approach to coding $\hat{a}$ and $\hat{c}$ and assign each a cost of $1/2\log n$ bits. Rissanen (1989) argued that no matter how one accounts for the hyperparameters, their contribution to the overall code length should be small. This reasoning is borne out in our simulation studies. At the end of this section we return to the issue of coding hyperparameters and discuss reasonable alternatives to the two-stage procedure motivated here.

An important ingredient in our code length (32) is the prior variance–covariance matrix, Σ. As mentioned earlier, for most values of Σ we cannot find a closed-form expression for $\hat{c}$ and instead must rely on an iterative scheme. (A general form for the procedure is outlined in the Appendix.) Rissanen (1989) gave details for the special case $\Sigma = I_{k\times k}$. We refer to the criterion derived under this specification as iMDL, where i refers to its use of the identity matrix. In the Bayesian literature on linear models, several authors have suggested a computationally attractive choice for Σ, namely $\Sigma = (X^tX)^{-1}$. Zellner (1986) christened this specification the g-prior. In our context, this value of Σ provides us with a closed-form expression for $\hat{c}$. After substituting $\hat{a} = R_c/n$ for a in (31), it is easy to see that

$$1/\hat{c} = \max(F - 1, 0) \quad \text{with} \quad F = \frac{(y'y - \text{RSS})}{kS}, \quad (33)$$

where F is the usual F ratio for testing the hypothesis that each element of β is 0, and $S = \text{RSS}/(n-k)$. The computations are spelled out in more detail in the Appendix. The truncation at 0 in (33) rules out negative values of the prior variance. Rewriting (33), we find that $\hat{c}$ is 0 unless $R^2 > k/n$, where R^2 is the usual squared multiple correlation coefficient. When the value of $\hat{c}$ is 0, the prior on β becomes a point mass at 0, effectively producing the "null" mixture model corresponding to $\gamma = (0, \ldots, 0)$. [The null model is a scale mixture of normals, each $N(0, \tau)$ and τ having an inverse-gamma prior.] Substituting the optimal value of $\hat{c}$ into (31) and adding

the cost to code the hyperparameters as in (32), we arrive at a final mixture form,

$$g\text{MDL} = \begin{cases} \frac{n}{2}\log S + \frac{k}{2}\log F + \log n, & \text{if } R^2 \geq k/n \\ \frac{n}{2}\log\left(\frac{y'y}{n}\right) + \frac{1}{2}\log n & \text{otherwise,} \end{cases} \tag{34}$$

which we call gMDL for its use of the g-prior. From this expression, we have dropped a single bit that is required to indicate whether the condition $R^2 < k/n$ is satisfied and hence which model was used to code the data. When $R^2 < k/n$, we apply the null model, which does not require communicating the hyperparameter $\hat{c}$. Hence a $1/2\log n$ term is missing from the lower expression.

Unlike most choices for Σ, the g-prior structure provides an explicit criterion that we can study theoretically. First, because $n/n = 1 \geq R^2$, this version of mixture MDL can never choose a model with dimension larger than the number of observations. A little algebra clearly shows that gMDL orders models of the same dimension according to RSS; that is, holding k fixed, the criterion (34) is an increasing function of RSS. This property is clearly shared by AIC, BIC, and C_p. Unlike these criteria, however, gMDL applies an adaptively determined penalty on model size. Rewriting (34) in the form

$$\frac{n}{2}\log\text{RSS} + \frac{\alpha}{2}k, \tag{35}$$

we find that α depends on the F-statistic, so that gMDL adapts to behave like AIC or BIC depending on which is more desirable (Hansen and Yu, 1999).

Finally, in Section 3.2 we applied a simple approximation to the mixture form of MDL to derive the so-called Stochastic Information Complexity (SIC) (17). For a model index γ, the Hessian matrix of the mixture $m(\cdot)$ in (12) based on the $k+1$ parameters β and $\tau = \sigma^2$ is given by

$$\begin{pmatrix} \frac{1}{\tau}X'X & 0 \\ 0 & \frac{n}{2\tau^2} \end{pmatrix}.$$

Therefore, a little algebra reveals SIC,

$$\text{SIC}(\gamma) = \frac{n-k-2}{2}\log\text{RSS} + \frac{k}{2}\log n + \frac{1}{2}\log\det[X'X], \tag{36}$$

where we have omitted an additive constant that is independent of model choice.

Normalized Maximum Likelihood. As mentioned in Section 3.4, the NML form of MDL (Barron et al. 1998; Rissanen 1996) is new, and only some of its theoretical properties are known. It is motivated by the ML code introduced by Shtarkov (1987). Recall that the ML estimates of β and $\tau = \sigma^2$ are given by (26). Let $f(y|X,\beta,\tau)$ be the joint Gaussian density of the observed data y, so that the NML function is

$$\hat{f}(y) = \frac{f(y|X,\hat{\beta}(y),\hat{\tau}(y))}{\int_{\mathcal{Y}(r,\tau_0)} f(z|X,\hat{\beta}(z),\hat{\tau}(z))\,dz}, \tag{37}$$

where $\mathcal{Y}(r,\tau_0) = \{z|\hat{\beta}'(z)X'X\hat{\beta}(z)/n \leq r, \hat{\tau}(z) \geq \tau_0\}$. In this case the maximized likelihood is not integrable, and our solution is to simply restrict the domain of $\hat{f}$ to $\mathcal{Y}$. Recall that we did not encounter this difficulty with the Bernoulli model studied in Section 3.4, where given the number of 1's, the binary sequences had a uniform distribution over the type class. Using the sufficiency and independence of $\hat{\beta}(y)$ and $\hat{\tau}(y)$, one obtains

$$-\log\hat{f}(y) = \frac{n}{2}\log\text{RSS} - \log\Gamma\left(\frac{n-k}{2}\right) - \log\Gamma\left(\frac{k}{2}\right) + \frac{k}{2}\log\frac{r}{\tau_0} - 2\log(2k). \tag{38}$$

To eliminate the hyperparameters r and τ_0, we again minimize the foregoing code length for each model by setting

$$\hat{r} = \frac{\hat{\beta}'(y)X'X\hat{\beta}(y)}{n} = \frac{y'y - \text{RSS}}{n} \quad \text{and} \quad \hat{\tau}_0 = \frac{\text{RSS}}{n}.$$

By substituting these values for r and τ_0 into (38), we obtain the selection criteria nMDL (n for normalized),

$$\text{nMDL} = \frac{n}{2}\log\text{RSS} - \log\Gamma\left(\frac{n-k}{2}\right) - \log\Gamma\left(\frac{k}{2}\right) + \frac{k}{2}\log\frac{y'y - \text{RSS}}{\text{RSS}} - 2\log(2k). \tag{39}$$

Technically, we should also add $1/2\log n$ for each of the optimized hyperparameters as we had done for gMDL. In this case the extra cost is common to all models and can be dropped. Rewriting this expression, we find that

$$\begin{aligned} \text{nMDL} = {} & \frac{n}{2}\log S + \frac{k}{2}\log F \\ & + \frac{n-k}{2}\log(n-k) - \log\Gamma\left(\frac{n-k}{2}\right) \\ & + \frac{k}{2}\log(k) - \log\Gamma\left(\frac{k}{2}\right) - 2\log k, \end{aligned}$$

up to an additive constant that is independent of k. Applying Stirling's approximation to each $\Gamma(\cdot)$ yields

$$\text{nMDL} \approx \frac{n}{2}\log S + \frac{k}{2}\log F + \frac{1}{2}\log(n-k) - \frac{3}{2}\log k.$$

We recognize the leading two terms in this expression as the value of gMDL (34) when $R^2 > k/n$. This structural similarity is interesting given that these two MDL forms were derived from very different codes.

Our derivation of nMDL follows Barron et al. (1998), who remedied the nonintegrability of the maximized likelihood by restricting $\hat{f}$ to the bounded region $\mathcal{Y}$. Recently, Rissanen (2000) addressed this problem by applying another level of normalization. Essentially, the idea is to treat the hyperparameters τ_0 and r as we did β and τ. The maximized likelihood (39) is normalized again, this time with respect to $\hat{\tau}_0 = \hat{\tau}_0(y)$ and $\hat{r} = \hat{r}(y)$. Following a straightforward conditioning argument, Rissanen (2000) found that this second normalization makes the effect of the hyperparameters on the resulting code length additive and hence can be ignored for model selection. [In deriving his form of NML, Rissanen (2000) also handled the issue of coding the model index γ differently than we have

in (24). Another normalization is applied, this time across a set of model indices Ω.] Ultimately, the final NML criterion derived in this way differs from our nMDL rule in (39) by only an extra $\log k$. Rissanen (2000) applied his NML selection criterion to wavelet denoising, illustrating its performance on a speech signal.

Stine and Foster (1999) also explored the derivation of NML for estimating the location parameter in a one-dimensional Gaussian family, but proposed a different solution to the nonintegrability problem. They suggested a numerically derived form that is shown to have a certain minimax optimality property (up to a constant factor). In general, the derivation of NML in such settings is still very much an area of active research. We present nMDL here mainly to illustrate the reasoning behind this form, and comment on its similarity to gMDL.

Discussion. As mentioned at the beginning of this section, there are alternatives to our use of the Bernoulli(1/2) model for coding the index γ. For example, George and Foster (2001) took the elements of γ to be a priori independent Bernoulli random variables with success probability p. They then selected a value for p by ML (in the same way that we treated the parameters a and c). In early applications of model selection to wavelet expansions, the value of p was fixed at some value less than 1/2 to encourage small models (Clyde, Parmigiani, and Vidakovic 1998).

The use of a normal inverse-gamma prior with $\Sigma = (X^t X)^{-1}$ appears several times in the literature in Bayesian model selection. For example, Akaike (1977) essentially derived gMDL for orthogonal designs, and Smith and Spiegelhalter (1980) used this prior when considering model selection based on Bayes factors where $a = 0$ and $c = c(n)$ is a deterministic function of sample size. These authors were motivated by a "calibration" between Bayes factors and penalized selection criteria in the form of BIC and AIC (see also Smith 1996; Smith and Kohn 1996). Finally, Peterson (1986) built on the work of Smith and Spiegelhalter (1980) by first choosing $\Sigma = (X^t X)^{-1}$ and then suggesting that c be estimated via (marginal) ML based on the same mixture (31). This is essentially Rissanen's (1989) prescription.

Throughout our development of the various MDL criteria, we have avoided the topic of estimating the coefficient vector β once the model has been selected. In the case of AIC and BIC, it is common practice to simply rely on OLS. But the resemblance of mixture MDL to Bayesian schemes suggests that for this form, a shrinkage estimator might be more natural. For example, the criterion gMDL is implicitly comparing models not based on $\hat{\beta}$, but rather on the posterior mean (conditional on our choice of model)

$$\max\left(1 - \frac{1}{F}, 0\right)\hat{\beta}$$

associated with the normal inverse-gamma prior and the regression model (23). Here F is defined as in the gMDL criterion (34). Recall that the condition that $F > 1$ is equivalent to the multiple R^2 being larger than k/n. Interestingly, this type of shrinkage estimator was studied by Sclove (1968) and Sclove, Morris, and Radhakrishnan (1972), who showed it to have improved mean squared error performance over OLS and other shrinkage estimators. In the case of iMDL, the coefficient vector β is estimated via classical ridge regression. Of course, Bayesian methods can be applied more generally within the MDL framework. For example, in Section 3.1 we found that any $\sqrt{n}$-consistent estimator can be used in the two-stage coding scheme. This means that we could even substitute Bayesian estimators for σ^2 and β in the two-stage criterion (28) rather than $\hat{\beta}$ and $\hat{\sigma}^2$. The beauty of MDL is that each such scheme can be compared objectively, regardless of its Bayesian or frequentist origins.

Next, in several places we are forced to deal with hyperparameters that need to be transmitted so that the decoder knows which model to use when reconstructing the data y. We have taken a two-stage approach, attaching a fixed cost of $1/2 \log n$ to each such parameter. Rissanen (1989) proposed using the universal prior on integers L^* after discretizing the range of the hyperparameters in a model-independent way. If prior knowledge suggests a particular distribution, then naturally it should be used instead. In general, the values of the hyperparameters are chosen to minimize the combined code length

$$(\hat{a}, \hat{c}) = \min_{(a,c)} \{L(y|X, a, c) + L(a) + L(c)\}, \tag{40}$$

where the first term represents the cost of coding the data given the value of the hyperparameters, $\hat{a}$ and $\hat{c}$, and the second term accounts for the overhead in sending them. In our derivation of iMDL and gMDL, we took the latter terms to be constant, so that we essentially selected the hyperparameters via ML (mixture or marginal). In the simulation study presented in the next section, each reasonable method for incorporating the cost of the hyperparameters produced selection criteria with similar prediction errors. As a final note, the theoretical material in Section 5 justifies the use of MDL *only* when the values of the hyperparameters are fixed. The minimization in (40) complicates a general analysis, but certainly selection rules can be studied on a case-by-case basis when explicit forms appear (as in the case of gMDL). We leave a detailed discussion of this material to future work.

4.1.2 A Simulation Study. When choosing between models with the same number of variables, AIC and each of the MDL procedures BIC, gMDL, and nMDL select the model with the smallest RSS. Therefore, to implement these criteria, it is sufficient to consider only the lowest-RSS models for dimensions $1, 2, \ldots, M$. When the number of predictors is relatively small (say, less than 30), it is not unreasonable to perform an exhaustive search for these models by a routine branch-and-bound algorithm (see Furnival and Wilson 1974 for a classic example). Unfortunately, the criteria iMDL and SIC involve characteristics of the design matrix X, requiring a different technique. An obvious (and popular) choice involves greedy, stepwise model building. In this case, some combination of stepwise addition (sequentially adding new variables that create the largest drop in the model selection criterion) and deletion (removing variables that have the least impact on the criterion) can be used to identify a reasonably good collection of predictors. Rissanen (1989) discussed these greedy algorithms in the context of (approximately) minimizing iMDL or SIC. The recent interest in Bayesian computing

has produced a number of powerful McMC schemes for variable selection. To apply these ideas to MDL, first recall that the mixture form is based on an integrated likelihood (12) that we can write as $m(y)=p(y|\gamma)$ for model indices γ. Assuming that each $\gamma\in\{0,1\}^M$ is equally likely a priori, we find that

$$m(y)=p(y|\gamma)\propto p(\gamma|y),$$

a posterior distribution over the collection of possible models. Candidate chains for exploring this space include the Gibbs sampler of George and McCulloch (1993); the importance sampler of Clyde, DeSimone, and Parmigiani (1996), applicable when the predictor variables are orthogonal; and the Occam's window scheme of Madigan, Raftery, and Hoeting (1997). In the simulation study described here, however, the number of covariates is small, so we can simply evaluate SIC and iMDL on *all* possible models to identify the best.

To understand the characteristics of each MDL criterion, we consider three simulated examples. These have been adapted from similar experiments of Tibshirani (1996) and Fourdrinier and Wells (1998). In each case, we work with datasets comprising 20 observations from a model of the form

$$y=x\beta+\sigma\epsilon, \tag{41}$$

where $x\in\mathbb{R}^8$ has a multivariate normal distribution with mean 0 and variance–covariance matrix $V_{ij}=2\rho^{|i-j|}$, $i,j=1,\ldots,8$, and ϵ is an independent standard normal noise term. Table 1 compares several MDL selection criteria across 100 datasets simulated according to (41), where $\rho=.5$, $\sigma=4$, and $\beta\in\mathbb{R}^8$ is assigned one of three (vector) values listed in Table 1. We quote both the average size of models selected by each criteria as well as the median model error, where model error is defined as

$$E\{x\hat{\beta}-x\beta\}^2=(\hat{\beta}-\beta)'V(\hat{\beta}-\beta),$$

with $\hat{\beta}$ obtained by an OLS fit with the selected variables. In Table 1 we also include the signal-to-noise (SNR) ratio for each set of simulations, where we take

$$\text{SNR}=\beta'V\beta/\sigma^2.$$

The row labeled OLS represents a straight OLS fit to the complete set of variables.

In this simulation, we initially compared AIC, BIC, gMDL, SIC, and nMDL. An anonymous referee suggested that as AIC is based on large-sample approximations, a modified criterion, AIC_C, is a more appropriate comparison. This form was derived by Sugiura (1978) for use in small samples and was later studied by Hurvich and Tsai (1989). In our notation this criterion is given by

$$\text{AIC}_\text{C}=\frac{n}{2}\log\text{RSS}+\frac{n}{2}\frac{1+k/n}{1-(k+2)/n}.$$

It is well known that when the data-generating mechanism is infinite-dimensional (and includes the candidate covariate variables), then AIC is an optimal selection rule in terms of prediction error; that is, AIC identifies a finite-dimensional model that, although an approximation to the truth, has good prediction properties. But, when the underlying model is in fact finite-dimensional (the truth belongs to one of the model classes being evaluated), AIC tends to choose models that are too large. The criterion AIC_C was derived under the assumption of a finite truth, and avoids the asymptotic arguments used in the original derivation of AIC. Computationally, this criterion is also amenable to the branch-and-bound techniques mentioned earlier.

In general, except for SIC, the MDL criteria outperformed AIC, AIC_C, and BIC. Note that AIC_C improves over AIC in all but the case of entirely weak effects, and even here the difference is small. This improvement is to be expected, because as the data-generating model is among the candidates being evaluated, precisely the finite-dimensional setup under which AIC_C was derived. The selection rule iMDL seems to perform exceedingly well in each simulation setup, although its performance degraded slightly when we considered larger sample sizes. In only one of the simulation suites did gMDL perform poorly relative to the other MDL schemes, namely the third case with entirely weak effects. When we increase the sample size to 50 but maintain the same SNR, gMDL recovers, and its model error rivals that of iMDL. Another interesting effect to mention in Table 1 is that in the third case (weak effects), model selection with iMDL outperforms OLS and AIC. In principle, AIC is known to work well in this situation. When we reran these simulations with $\rho=0$, corresponding to independent predictors, AIC did in fact improve to the level of iMDL. The implicit shrinkage performed by iMDL when evaluating models through (32) is apparently responsible for iMDL's excellent performance here. We hasten to add, however, that in all cases, once a model is selected, we are simply performing an OLS fit to obtain $\hat{\beta}$ (from which the model error is derived). For both mixture forms of MDL and for all of the simulations, the shrinkage procedures based on $\hat{c}$ improve on these OLS estimates.

Given the penalties on k imposed by AIC and BIC, one can expect AIC to favor larger models and BIC to be more conservative. This can be seen in each of our simulation results. But the MDL forms can be thought of as imposing an adaptive penalty on model size. For comparison purposes, we computed an *equivalent* penalty in a neighborhood of the best model identified by the MDL criteria. To be more precise, Figure 5 plots the iMDL criterion versus model size, evaluated for the $2^8=512$ possible models using data from a single run of the simulation described earlier. Define $\text{iMDL}^*(k)$ to be the minimum value of iMDL among all models of size k and let $\text{RSS}^*(k)$ be the residual sum of squares for that model. Then consider the quantity

$$\lambda(k)=2\left[\text{iMDL}^*(k)-\frac{n}{2}\log\text{RSS}^*(k)\right].$$

If we replaced iMDL with either AIC or BIC in this definition, then the difference $\lambda(k+1)-\lambda(k)$ would be 2 or $\log n$. (Although the expressions for AIC and BIC can be manipulated in other ways to tease out the penalty on dimension, we have chosen differences because most of the MDL expressions are only known up to additive constants.) To get a rough idea of the price placed on dimension by the MDL criteria, we looked at this difference in the neighborhood of the minimum. In Figure 5, the heavy black line joins the two points

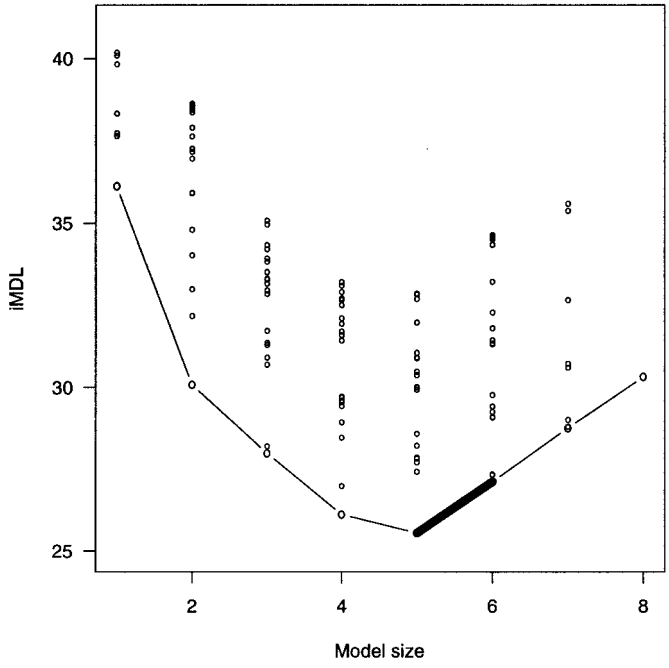

Figure 5. Calculating an Equivalent Penalty for the MDL Criteria. In this case we consider iMDL *and restrict our attention to a difference of the two points connected by heavy black segments.*

used to evaluate $\lambda(k)$. The average equivalent penalty across the 100 replicates of each simulation is given in Table 1. The adaptability of these procedures is immediately evident from the first and third simulation setups. When faced with a single, strong effect, for example, the penalties associated with iMDL and gMDL are larger than that of BIC, forcing smaller models; whereas when given a number of small effects, the penalty shrinks below that for AIC allowing iMDL to capture larger models. SIC tends to impose a penalty that is much weaker than AIC, leading to its discouraging results.

These simulations demonstrate a distinct performance advantage in the adaptive forms of MDL, *g*MDL and iMDL, over BIC, AIC, and AIC_C in model selection. The theoretical properties of gMDL and iMDL are currently under study (Hansen and Yu 1999). Interestingly, both of these forms share much in common with the new empirical Bayes criteria of George and Foster (1998) and the *Peel* method of Fourdrinier and Wells (1998). In the next section we investigate the use of MDL in two applied problems. In the first case, we propose a hand-crafted procedure to perform model selection within a restricted class of problems. We find that the adaptivity of MDL produces results that are (automatically) equivalent to this specialized approach. In the second example, we apply MDL to curve estimation. We use the output from this procedure later to illustrate a form of MDL for cluster analysis.

4.1.3 Applying Minimum Description Length in Practice: Two Regression Examples

The Genetics of a Fruit Fly. Our first example comes from genetics and has been developed into a variable selection problem by Cowen (1989), Doerge and Churchill (1996), and Broman (1997). The data we consider were collected by Long et al. (1995) as part of an experiment to identify *genetic loci*, locations on chromosomes, that influence the number of bristles on the fruit fly *Drosophila melanogaster*.

The experimental procedure followed by Long et al. (1995) was somewhat complicated, but we attempt to distill the essential features. First, a sample of fruit flies was selectively inbred to produce two family lines differentiated on the basis of their abdominal bristles. Those flies with low bristle counts

Table 1. Simulation Results for $n=20$ Observations From (41). In each case, $\rho=.5$ and $\sigma=4$.

	Criterion	Median model error	Average model size	Proportion correct	Equivalent penalty
$\beta=(5,0,0,0,0,0,0,0)$	OLS	9.1	8.0	0	0
(SNR $\approx$ 3.2)	*g*MDL	1.0	1.4	.7	4.0
	nMDL	4.2	2.3	.2	2.4
	iMDL	1.4	1.5	.6	3.7
	BIC	3.2	1.9	.4	3.0
	AIC	5.3	2.8	.2	2.0
	AIC_C	3.3	1.9	.4	3.2
	SIC	7.6	4.1	.04	1.0
$\beta=(3,1.5,0,0,2,0,0,0)$	OLS	9.6	8.0	0	0
(SNR $\approx$ 3.2)	gMDL	7.6	2.8	.2	3.6
	nMDL	7.6	3.5	.3	2.6
	iMDL	6.8	3.0	.3	2.7
	BIC	8.0	3.3	.2	3.0
	AIC	8.5	3.8	.2	2.0
	AIC_C	7.6	3.0	.3	3.6
	SIC	8.6	5.1	.07	1.0
$\beta=0.75*(1,1,1,1,1,1,1,1)$	OLS	9.5	8.0	1.0	0
(SNR $\approx$ 1.4)	gMDL	10.5	2.9	.0	2.9
	nMDL	9.7	3.6	.0	1.8
	iMDL	9.3	3.4	.0	1.9
	BIC	11.0	3.0	.0	3.0
	AIC	10.2	3.5	.0	2.0
	AIC_C	10.6	2.8	.0	3.5
	SIC	10.5	4.8	.06	1.0

was separated into one parental line L, whereas those with high counts formed another line, H. Several generations of flies were then obtained from these two populations through a *backcross*. That is, the H and L lines were crossed to yield the so-called first filial generation F_1, and then the F_1 flies were again crossed with the low parental line L. Ultimately, 66 inbred family lines were obtained in this way, so that the individual flies within each group were genetically identical at 19 chosen genetic markers (or known locations on the chromosomes). Abdominal bristle counts were collected from a sample of 20 males and 20 females from each of these populations. By design, all of the flies bred in the backcross inherited one chromosome from the first filial generation F_1 and one chromosome from the low parental line L, so that at each of the genetic markers they had either the LL or HL genotype. The goal of this experiment was to identify whether the genotype at any of the 19 genetic markers influenced observed abdominal bristle counts.

Let y_{ij}, $i=1,\ldots,66$, $j=1,2$, denote the average number of bristles for line i, tabulated separately for males, corresponding to $j=1$, and females, corresponding to $j=2$. Consider a model of the form

$$y_{ij}=\mu+\alpha s_j+\sum_l \beta_l x_{il}+\sum_l \delta_l s_j x_{il}+\epsilon_{ij}, \tag{42}$$

where s_j is a contrast for sex, $s_1=-1$ and $s_2=+1$; and $x_{il}=-1$ or $+1$ according to whether line i had genotype LL or HL at the lth marker, $l=1,\ldots,19$. Thus the full model (42) includes main effects for sex and genotype as well as the complete sex $\times$ genotype interaction, a total of 39 variables. The error term ϵ_{ij} is taken to be Gaussian with mean 0 and unknown variance σ^2. In this framework identifying genetic markers that influence bristle counts becomes a problem of selecting genotype contrasts in (42). Following Broman (1997), we do not impose any hierarchical constraints on our choice of models, so that any collection of main effects and interactions can be considered. Thus, in the notation of Section 4.1 we introduce an index vector $\gamma\in\{0,1\}^{39}$ that determines which covariates in (42) are active. (We have intentionally excluded the intercept from this index, forcing it to be in each model.)

Broman (1997) considered variable selection for this problem with a modified BIC criterion,

$$\mathrm{BIC}_\eta=\frac{n}{2}\log\mathrm{RSS}+\eta\frac{k}{2}\log n, \tag{43}$$

where $\eta=2$, 2.5, or 3. Broman (1997) found that placing a greater weight on the dimension penalty $\log(n)/2$ is necessary in this context to avoid including spurious markers. As with the data from Long et al. (1995), model selection is complicated by the fact that the number of cases n collected for backcross experiments is typically a modest multiple of the number of possible predictor variables. Aside from practical considerations, Broman (1997) motivated (43) by appealing to the framework of Smith (1996) and Smith and Kohn (1996). These authors started with the mixture distribution (31) derived in Section 4.1.1, taking the improper prior specification $a=d=0$ in (A.2) and (A.3). Instead of finding optimal values for c, they considered deterministic functions $c=c(n)$. This approach was also taken by Smith and Spiegelhalter (1980), who attempted to calibrate Bayesian analyses with other selection criteria AIC. If we set $c(n)=n^\eta$ for all models, then from (31) we roughly obtain Broman's criterion (43). (This argument is meant as a heuristic; for the precise derivation of (43), see Broman 1997.) The larger we make η, the more diffuse our prior on β becomes. Because the same scaling factor appears in the prior specification for models of different dimensions, the mass in the posterior distribution tends to concentrate on models with fewer terms.

Because the number of markers studied by Long et al. (1997) was relatively small, Broman (1997) was able to use a branch-and-bound procedure to obtain the optimal model according to each of the criteria (43). By good fortune, these three rules each selected the same eight-term model,

$$y_{ij}=\mu+\alpha s_j+\beta_2 x_{i2}+\beta_5 x_{i5}+\beta_9 x_{i9}+\beta_{13} x_{i,13}+\beta_{17} x_{i,17}+\delta_5 s_j x_{i5}+\epsilon_{ij}, \tag{44}$$

which includes the main effect for sex, five genotype main effects (occurring at markers 2, 5, 9, 13, and 17), and one sex $\times$ genotype interaction (at marker 5). To make a comparison with the MDL selection rules derived in Section 4.1.1, we again performed an exhaustive search for AIC, BIC, gMDL, and nMDL. As noted earlier, a number of McMC schemes can be applied to find promising models based on iMDL and SIC. We chose the so-called focused sampler of Wong, Hansen, Kohn, and Smith (1998). (The specific sampler is somewhat unimportant for the purpose of this article. Any one of a number of schemes could be used to accomplish the same end.) In Figure 6 we overlay these criteria, plotting the minimum of each as a function of the model dimension k. For easy comparison, we mapped each curve to the interval $[0,1]$. As noted by Broman (1997), BIC and hence also AIC chose larger models that were primarily supersets of (44) involving 9 and 13 terms. Our two forms of mixture MDL, gMDL and iMDL, and the NML criterion, nMDL, were each in agreement with Broman's BIC_η, selecting (44). Using the device introduced in the previous section (see Fig. 4), we find that the equivalent penalty imposed by gMDL was 7.4, which corresponds to an $\eta=7.4/\log n=7.4/\log 132=1.5$. For nMDL the story was about the same, with an equivalent penalty of 7.0 (or an η of 1.4). Finally, iMDL had a penalty of 6.4 for an η of 1.3. These findings are satisfying in that our automatic procedures produced the same results as selection rules that have been optimized for the task of identifying nonspurious genetic markers from backcross experiments. Somewhat disappointingly, strict minimization of SIC identifies a model with 12 variables (and an equivalent penalty of 1.6, less than half of BIC's $\log 132=4.9$). From Figure 5, however, we see that SIC curve is extremely flat in the neighborhood of its optimum, implying that an 11-term model provides virtually the same quality of fit. For $k=11$, SIC selects a model that is a subset of that chosen according to AIC, but contains all of the terms in the model identified by BIC.

To summarize, we have compared the performance of several forms of MDL to a special-purpose selection criterion

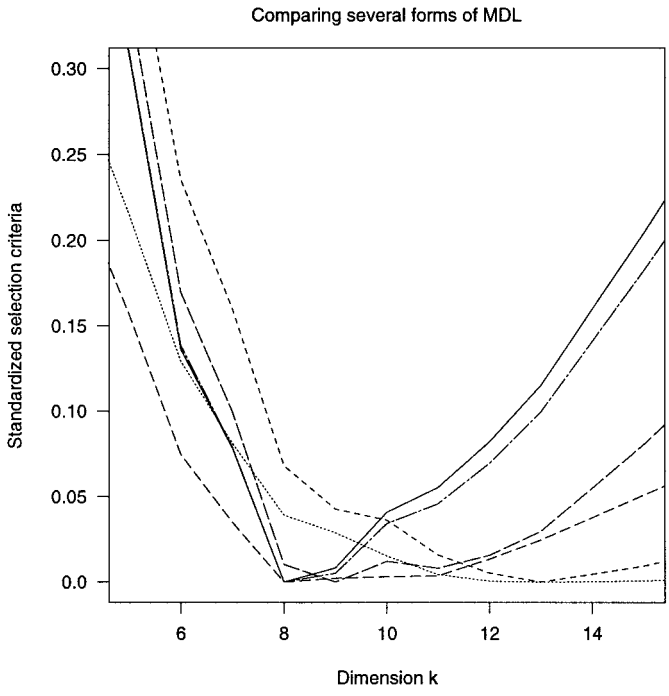

Figure 6. Comparing Several Different Model Selection Criteria: gMDL, (——), nMDL (–·–), BIC (– –), iMDL (– – ·), AIC (- - - ·), SIC (·····).

(43). For the most part, our results are consistent with those of Broman (1997), identifying (44) as the best model. The only poor performer in this context was SIC, which fell between the poorly performing criteria AIC and BIC.

The Color of Supermarket Produce. Our second regression example involves model selection in the context of function estimation. Figure 7 presents a number of *spectral reflectance curves* obtained from samples of common fruits and vegetables. Measurements were taken on samples from some 70 varieties of popular produce, with the ultimate goal of creating a recognition system that could augment supermarket checkout systems. For example, in Figure 6(a), each curve represents the color of a lemon measured at a small spot on its surface. The intensity of light reflected by its skin is recorded as a function of wavelength, producing a single curve in Figure 7. Because of noise considerations, we have restricted our measurements to a subset of the visible spectrum between 400 and 800 nm, recording values in 5-nm intervals. To remove the effects of varying surface reflectivity and to account for the possibility that the intensity of the incident light may vary from measurement to measurement, each curve has been normalized (across wavelength) to have mean 0 and variance 1.

To make sense of these curves, consider the sample of limes represented in Figure 6(c). Limes are green because chlorophyll in their skin absorbs light strongly in the region between 680 and 700 nm. The dip in this region is evident in each of the lime measurements. Similarly, several of the bananas in our sample must have been slightly green, because a few of the corresponding curves also drop in this region. In general, plant pigments absorb light in broad, overlapping bands, and hence we expect our reflectance curves to be smooth functions of wavelength. The underlying chemistry manifests itself by varying the coarse features of each measurement. Finally,

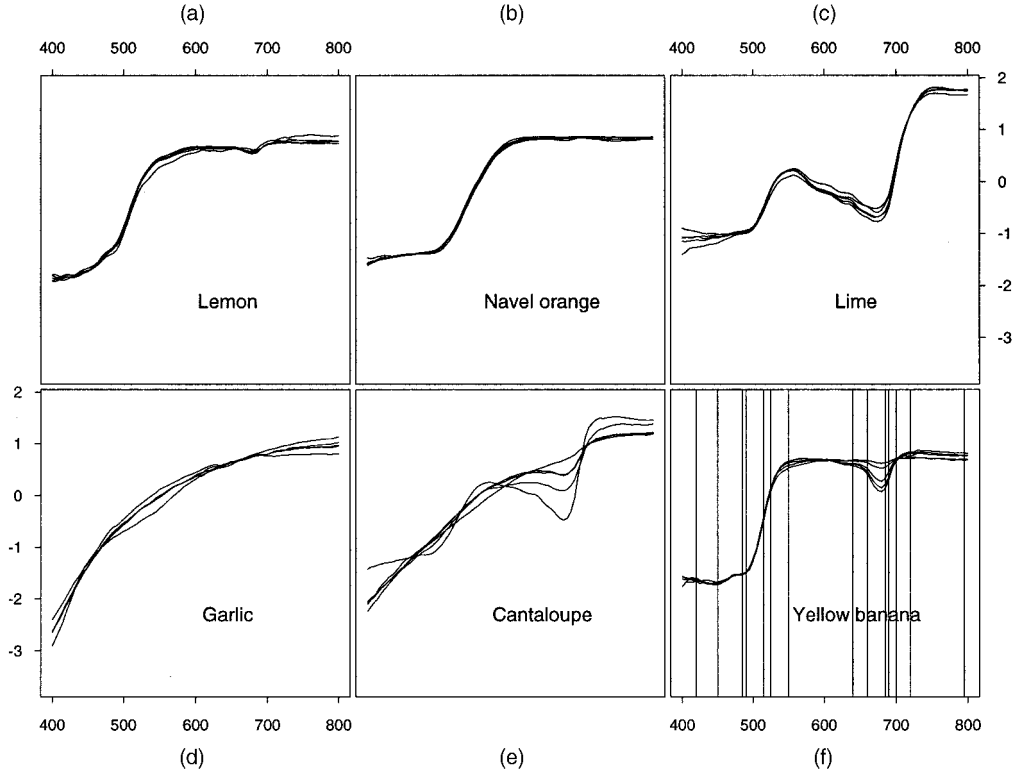

Figure 7. Spectral Reflectance Curves Collected From Six Varieties of Supermarket Produce. In (a)–(f) we plot five representative curves. Knot locations selected by gMDL *and* BIC *are marked by vertical lines in (f).*

as should be apparent from Figure 6, our experimental setup allowed us to capture these curves with very little noise.

In this section our goal is to derive a compact representation of these curves to be used for recognition purposes (see also Furby, Kiiveri, and Campbell 1990). Dimension reduction is accomplished by simple projections onto an adaptively determined space of functions. Suppose that we observe each curve at n distinct wavelengths $x_1, \ldots, x_n$. Then consider the candidate basis functions of the form

$$B_i(x) = K(x, x_i) \quad \text{for } i = 1, \ldots, n,$$

where $K(\cdot, \cdot)$ is some specified *kernel* function. There are a number of choices for K, most falling into the class of so-called *radial basis functions* often used in neural networks (Hertz, Krough, and Palmer 1991). We choose instead to use the kernels that appear in the construction of smoothing splines (Wahba 1990; Wong, Hansen, Kohn, and Smith 1997). Then, having settled on a basis, we search for an approximation of the form

$$f(x) \approx \alpha_0 + \alpha_1 x + \sum_{i:\gamma_i=1} \beta_i B_i(x), \qquad x \in [400, 800], \qquad (45)$$

where f is the true reflectance measurement taken from a sample of fruit and $\gamma \in \{0,1\}^n$ again indexes the candidate basis functions. Variable selection in (45) with B_i defined through smoothing spline kernels is equivalent to choosing knot locations in a natural spline space (Schumaker 1993). Note that in this case we always include a constant term and a linear term in our fits. (Because of our normalization, we do not need the constant term, but we include it in the equation for completeness.) In this context, Luo and Wahba (1997) used a stepwise greedy algorithm to identify a model, and Wong et al. (1997) used the focused sampler after constructing a computationally feasible prior on γ. Finally, recall that a traditional smoothing spline estimate would fix $\gamma = (1, \ldots, 1)$ and perform a penalized fit (Wahba 1990). Hansen and Kooperberg (1998) gave a general discussion of knot location strategies.

As mentioned earlier, the data presented in Figure 7 were collected as part of a larger project to create a classifier for recognizing supermarket produce based solely on its color. Although we ultimately applied a variant of penalized discriminant analysis (Hastie, Buja, and Tibshirani 1995), a reasonably accurate scheme involves dimension reduction (45) followed by simple linear discriminant analysis (LDA) on the coefficients β_i. Therefore, we adapted the MDL criteria to handle multiple responses (curves). Our search for promising indices γ now represents identifying a single spline space (45) into which each curve is projected, producing inputs (coefficients) for a classification scheme like LDA. Given our extension of the MDL procedures to multiple responses, it is also possible to simply "plug in" each of these schemes to the flexible discriminant analysis technique of Hastie, Tibshirani, and Buja (1994). The expansion (45), with its curve-by-curve projection into a fixed linear (although adaptively selected) space, can be applied directly in this algorithm.

For our present purposes, we have roughly 30 curves for each variety shown in Figure 7, for a total of 176 response vectors. Because of the size of the problem, the best BIC and gMDL models were computed using the focused sampler of Wong et al. (1997). We restricted our attention to these two forms purely on the basis of computational burden. The iterations (66) required by iMDL are prohibitive given our current implementation of the algorithm. It is, of course, possible to take shortcuts with greedy, deterministic searches as proposed by Rissanen (1989). But to simplify our presentation, we restrict our attention to only these two forms. In each case, we used 10,000 iterations of the sampler to identify the best expansion (45). To simplify our exposition even further, we were pleased to find that BIC and gMDL agreed on the number of knots, and hence their placement as both select the minimal RSS model among candidates of the same dimension. Figure 7 highlights the locations of the selected knots, or rather the points x_i that correspond to kernel functions $B_i(\cdot) = K(\cdot, x_i)$ in the approximation (45). The higher density of knots in the neighborhood of 700 nm is expected. Because of chlorophyll's absorption properties, reflectance curves collected from green plants often exhibit a sharp rise in this region, known as the *red edge*.

Based on these selected knot locations, we now project each curve into the linear space defined in (45). In the next section we apply the coefficients from these projections to a MDL-like clustering scheme.

4.2 Clustering Analysis

In this section we apply a close cousin of MDL introduced by Wallace and Boulton (1968) and refined by Wallace and Freeman (1987). Originally designed for cluster analysis, their principle of minimum message length (MML) also appeals to a notion of code length to strike a balance between model complexity and fidelity to the data. Under this framework, a two-part message is constructed, analogous to the two-stage coding scheme discussed in Sections 2 and 3. For cluster analysis, a mixture of parametric models is proposed, so that the first part of the MML message consists of

- the number of clusters or components
- the number of data points belonging to each cluster
- the parameters needed to specify each model
- the cluster membership for each data point.

In the second part of the message, the data are encoded using the distribution of the specified model exactly as described in Sections 2 and 3. As with MDL, the best MML model is the one with the shortest message length. In the words of Wallace and Boulton (1968), "a classification is regarded as a method of economical statistical encoding of the available attribute information."

When possible, MML attempts to divide the data into homogeneous groups (implying that the model for each component captures the structure in the data), while penalizing the overall complexity or rather the total number of components. For the moment, the only practical difference between two-stage MDL and MML involves the precise encoding of the selected model. (As these details are somewhat technical, the interested reader is referred to Baxter and Oliver 1995.) Observe, however, that the restriction to two-part messages limits MML from taking advantage of other, more elaborate coding schemes that still give rise to statistically sound selection schemes.

To illustrate MML or the practical application of MDL to cluster analysis, we consider the produce data from the previous section. Recall that each spectral reflectance curve was projected onto a spline space (45) with the 14 knot locations specified in Figure 7. When combined with the linear term in (45), we obtain 15 estimated coefficients for each of our 176 curves. To this dataset we applied MML cluster analysis using SNOB, a public-domain Fortran program developed by Wallace's group at Monash University in Melbourne, Australia. (The SNOB program and a number of relevant documents can be found through David Dowe's Web site http://www.cs.monash.edu.au/~dld.) Wallace and Dowe (1994) have described the mixture modeling framework on which SNOB is based.

When clustering Gaussian data, each component of the mixture has a multivariate normal distribution with a diagonal covariance matrix. At present, SNOB assumes that all intraclass correlations are 0. Following a suggestion in the documentation, we orthogonalized the entire dataset via a principal components decomposition. Figure 8 plots the scores corresponding to the first two components, labeling points according to the class of each fruit. Clear divisions can be seen between, say, the limes and bananas. The cantaloupe measurements stretch across a broad area at the bottom of this plot, an indication that it will be difficult to separate this class from the others. This is perhaps not surprising given the different colors that a cantaloupe can exhibit. The 10-cluster SNOB model is superimposed by projecting each Gaussian density in the mixture onto the space of the first two-dimensional principal components. Again, each component in this mixture is a Gaussian with diagonal variance–covariance matrix. In some cases the SNOB clusters capture isolated groups of fruits (the bananas, lemons and limes, for example), whereas in other cases the color appears in too many different varieties.

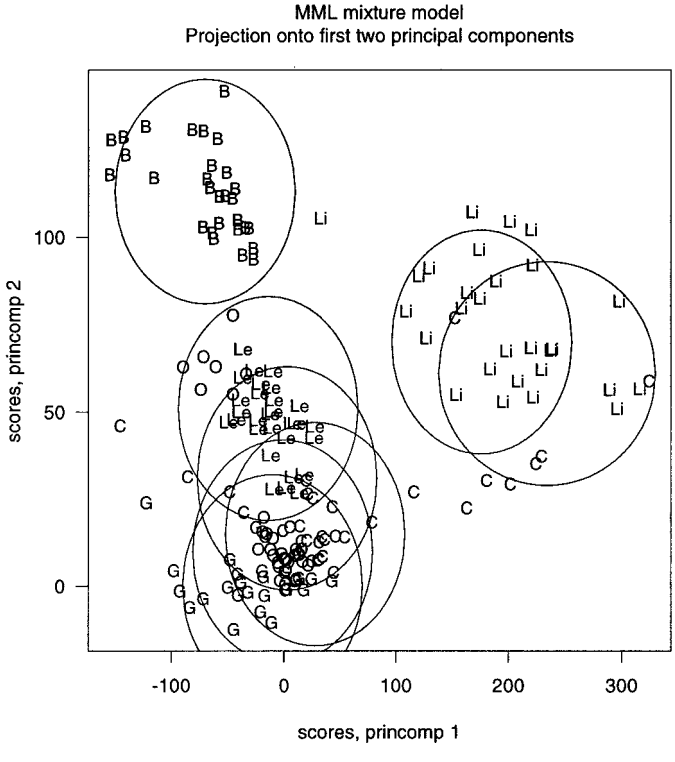

Figure 8. Mixture Modeling via MML. SNOB finds 10 clusters for the projected reflectance curves. The ovals are contours of constant probability for the clusters that exhibit significant variation in the first two principal component directions. B = Banana, Li = Lime, Le = Lemon, C = Cantaloupe, O = Orange, G = Garlic.

4.3 Time Series Models

Our final application of MDL is to time series analysis. We emphasize predictive MDL, which is especially natural in this setting. Our benchmarks will be AIC and BIC. In this context, determining the orders of an ARMA process is a common model selection problem. Throughout this section we focus on Gaussian ARMA(p,q) models, specified by the equation

$$x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + Z_t + \theta_1 Z_{t-1} \cdots + \theta_q Z_{t-q}, \qquad (46)$$

where the variables Z_t are iid Gaussian with mean 0 and variance σ^2. As is customary, we assume that the polynomials

$$1 - \phi_1 z - \cdots - \phi_p z^p = 0 \qquad \text{and} \qquad 1 - \theta_1 z - \cdots - \theta_q z^q = 0$$

have no roots in $|z| < 1$, so that (46) describes a stationary, second-order Gaussian process.

Given parameter values $\phi = (\phi_1, \ldots, \phi_p)$ and $\theta = (\theta_1, \ldots, \theta_q)$, and a series $x_1, \ldots, x_t$, it is straightforward to make predictions from (46) to times $t+1, t+2, \ldots$ conditional on the first t data points. For example, following Brockwell and Davis (1991, p. 256), x_{t+1} has a Gaussian distribution with mean $\hat{x}_{t+1}$ and variance $\sigma^2 r_t$, which are calculable from the recursive formulas

$$\begin{cases} \hat{x}_{t+1} = \sum_{i=1}^{t} \theta_{it}(x_{t+1-i} - \hat{x}_{t+1-i}), & 1 \le t < \max(p,q) \\ \hat{x}_{t+1} = \phi_1 x_t + \ldots + \phi_p x_{t+1-p} & \\ \qquad + \sum_{i=1}^{q} \theta_{it}(x_{t+1-i} - \hat{x}_{t+1-i}), & t \ge \max(p,q) \end{cases} \qquad (47)$$

The extra parameters θ_{it} and r_t can be obtained recursively by applying the so-called innovation algorithm (Brockwell and Davis, 1991, prop. 5.2.2.) to the covariance function of the ARMA process.

We now turn to defining two forms of MDL in this context. For ease of notation, we collect the parameters ϕ, θ, and σ^2 into a single vector β. To emphasize the dependence of $\hat{x}_{t+1}$ and r_t on β, we write

$$\hat{x}_{t+1}(\beta) \qquad \text{and} \qquad r_t(\beta).$$

Hence the predictive density of x_{t+1} conditional on $x_1, \ldots, x_t$ is given by

$$q_t(x_{t+1}|\beta) = \left(2\pi r_t \sigma^2\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2 r_t \sigma^2}(x_{t+1} - \hat{x}_{t+1})^2\right),$$

and the likelihood for β based on $x_1, \ldots, x_n$ is simply

$$q(\beta) = \prod_{t}^{n} q_t(x_{t+1}|\beta). \qquad (48)$$

Letting $\hat{\beta}_n$ denote the MLE in this context, two-stage MDL takes on the now familiar form of BIC,

$$-\log q(\hat{\beta}_n) + \frac{p+q+1}{2}\log n.$$

The consistency proof of the two-stage MDL or BIC follows from Hannan and Quinn (1979) for autoregressive models and from Gerencsér (1987) for general ARMA processes. As explained earlier, the complexity penalty $\log n/2$ comes from coding the parameter values at the estimation rate $1/\sqrt{n}$. Huang (1990) showed that when an AR model is not stable, this complexity penalty should be adjusted to the new estimation rate. For example, this leads to a complexity term $\log n$ for the explosive case where the estimation rate is $1/n$.

When modeling time series data, the predictive form of MDL is perhaps the most natural. Expressing the likelihood predictively, we arrive at the criterion

$$\mathrm{PMDL}(p,q) = -\sum_{t=1}^{n} \log q_t(x_{t+1}|\hat{\beta}_t). \tag{49}$$

A closely related quantity for assessing the orders in ARMA models is the so-called accumulated prediction error (APE),

$$\mathrm{APE}(p,q) = \sum_{t}^{n} (x_{t+1} - \hat{x}_{t+1})^2,$$

although APE was used long before the MDL principle. The computational cost of PMDL can be enormous for general ARMA models, because the parameter estimate $\hat{\beta}_t$ in (49) must be updated for each new observation. Hannan and Rissanen (1982) and Lai and Lee (1997) have proposed methods for reducing this cost. Consistency proofs for PMDL order selection were given for AR models by Hannan et al. (1988) and Hemerly and Davis (1989), and for general ARMA models by Gerencsér (1987).

Although deriving a mixture form of MDL appears possible by appealing to the state-space approach to ARMA processes (Carlin, Polson, and Stoffer 1992), selecting (computationally feasible) priors remains an active research area in its own right. In the next example, we apply AIC, BIC, and PMDL to the actual values (differenced) of the return series studied in Section 2.

Example 3 (Continued). In Figure 2(a) we presented first differences of the daily return series. Although our interest at that point was on compressing the string of ups and downs, we now focus on the series itself. To ease the computational burden of PMDL, we chose to update the parameter estimates only every 100 days. We also restricted our attention to the first 6,100 data points, intentionally stopping short of the spike induced by the stock market crash in 1987. Using the time series tools in S-PLUS, we fit our parameter estimates and recursively evaluated the likelihood (48) conditioned on the first 100 days. The standard analysis tools in S-PLUS allowed for a quick order determination via AIC and BIC. These criteria indicated that a simple MA(1) was in order. We then considered models where p and q varied (independently) over the range 0–5, and found that PMDL also favors a MA(1) model. This result agrees with our initial work on the up-and-down series from Section 2. Undoubtedly, the (twice-differenced) DJIA series is much more complex than a simple ARMA process, but our goal here is to illustrate the application of MDL, not to dabble in the stock market.

5. THEORETICAL RESULTS ON MINIMUM DESCRIPTION LENGTH

In Section 3 we mentioned that the validity of an MDL model selection criterion depends on properties of the underlying coding scheme or, more precisely, the resulting description lengths. In this section we formalize these ideas in the context of regular parametric families (model classes). We first derive pointwise and minimax lower bounds on the code length with which data strings can be encoded with the help of a class of models. Coding schemes yielding description lengths that achieve these lower bounds are said to produce valid MDL model selection criteria. Next, we return to the hypothesis tests of Example 4 and verify that the two-stage, predictive, and mixture forms of description length all achieve these lower bounds. It has been shown that under very general conditions, MDL model selection criteria are consistent when the data-generating model belongs to the class being considered (Barron et al. 1998). We end this section by illustrating why this is the case, using the same simple framework of Example 4. For a more thorough treatment of the theoretical justifications of MDL, the interested reader is referred to Barron et al. (1998).

5.1 Rissanen's Pointwise Lower Bound

Given a parametric family or model class

$$\mathcal{M} = \{f_\theta(x^n) : \theta \in \Theta \subset \mathbb{R}^k\},$$

let $E_\theta\{\cdot\}$ denote the expectation with respect to a random variable (data string) X^n with density f_θ. (In contrast to previous sections, here we are more careful when referring to random variables X^n versus points $x^n \in \mathbb{R}^n$.) Using this notation, the differential entropy of f_θ defined in (5) becomes

$$H_\theta(X^n) = -E_\theta \log f_\theta(X^n).$$

For any density (or prefix code) $q(x^n)$, the *Kullback–Leibler divergence* between f_θ and q is given by

$$\begin{aligned} R_n(f_\theta, q) &= E_\theta \log \frac{f_\theta(X^n)}{q(X^n)} \\ &= E_\theta\{-\log q(X^n) - [-\log f_\theta(X^n)]\}. \end{aligned} \tag{50}$$

Here $R_n(f_\theta, q)$ represents the expected extra nats needed to encode the data string X^n using q rather than the optimal scheme based on f_θ. In coding theory, R_n is called the (*expected*) *redundancy* of q.

Defining a valid description length for a data string based on models from the class $\mathcal{M}$ reduces to finding a density q that achieves the "smallest" redundancy possible for all members in $\mathcal{M}$. To make this concrete, we first derive a lower bound on redundancy in a well-defined global sense over the entire class $\mathcal{M}$, and then illustrate choices for q that achieve it. We begin with a pointwise result first derived by Rissanen (1986a).

Assume that a $\sqrt{n}$-rate estimator $\hat{\theta}(x^n)$ for θ exists and that the distribution of $\hat{\theta}(X^n)$ has uniformly summable tail probabilities,

$$P_\theta\{\sqrt{n}\|\hat{\theta}(X^n) - \theta\| \geq \log n\} \leq \delta_n, \quad \text{for all } \theta \text{ and } \sum_n \delta_n < \infty,$$

where $\|\theta\|$ denotes some norm in $\mathbb{R}^k$. Then for any density q, Rissanen (1986a) found that

$$\liminf_{n\to\infty} \frac{E_\theta \log[f_\theta(X^n)/q(X^n)]}{(k/2)\log n} \geq 1, \tag{51}$$

for all $\theta \in \Theta$, except on a set of θ with a Lebesgue measure 0. This exceptional set depends on q and k. Viewing $-\log q(X^n)$ as the code length of an idealized prefix code, (51) implies that without knowing the true distribution f_θ, we generally need at least $k\log n/2$ more bits to encode X^n, no matter what prefix code we use.

Shannon's Source Coding Theorem (Sec. 2) quantifies the best expected code length when symbols from a known data-generating source are encoded with the density q (denoted by the distribution function Q in Sec. 2). Rissanen's lower bound (51) extends this result to the case in which we only know that the "true" source belongs to some model class $\mathcal{Q}$. In coding theory this is referred to as the problem of *universal coding*. Historically, the pointwise lower bound was the first to appear, followed by the minimax approach discussed in the next section. The two approaches were connected by Merhav and Feder (1995), who obtained a lower bound on redundancy for abstract spaces. The pointwise lower bound (51), has been generalized to a special nonparametric class of models in density estimation by Rissanen, Speed, and Yu (1992), and their arguments should apply to other nonparametric settings.

5.2 Minimax Lower Bound

The bound (51) holds for almost every value of $\theta \in \Theta$—hence the term pointwise. We now turn to a minimax version of this result, again focusing on parametric classes. (The interested reader is referred to Barron et al. (1998) for the minimax approach in MDL and nonparametric estimation.)

First, we define the minimax redundancy to be

$$R_n^+ = \min_q \sup_{\theta\in\Theta} R_n(f_\theta, q). \tag{52}$$

This expression has a simple interpretation as the minimum over all coding schemes for X^n of the worst-case redundancy over all parameter values θ. Next, consider a prior distribution $w(\theta)$ on the parameter space Θ and define the *Bayes redundancy* associated with a density q relative to w as

$$R_n^*(q, w) = \int_\Theta R_n(f_\theta, q) w(d\theta). \tag{53}$$

The *minimal Bayes redundancy* for a given w is given by

$$R_n(w) = \min_q R_n^*(q, w), \tag{54}$$

which is achieved by the mixture distribution

$$m^w(x^n) = \int_\Theta f_\theta(x^n) w(d\theta). \tag{55}$$

To see this, write

$$R_n^*(q, w) - R_n^*(m^w, w) = \int_{\mathcal{X}^n} \log \frac{m^w(x^n)}{q(x^n)} m^w(dx^n) \geq 0,$$

where the last relation holds from Jensen's inequality. Evaluating (54) at m^w yields

$$\begin{aligned} R_n(w) &= R_n^*(m^w, w) \\ &= \int_\Theta \int_{\mathcal{X}^n} \log \frac{f_\theta(x^n)}{m^w(x^n)} f_\theta(dx^n)\, w(d\theta). \end{aligned}$$

With a slight abuse of notation, if we let Θ also denote the random variable induced by the prior w, then the preceding expression is known as the mutual information $I_w(\Theta; X^n)$ between Θ and the random variable $X^n = X_1, \ldots, X_n$ (Cover and Thomas 1991). Thus we have established that

$$R_n(w) = I_w(\Theta; X^n). \tag{56}$$

The quantity I_w measures the average amount of information contained in the data X^n about the parameter Θ and was used to measure information in a statistical context by Lindley (1956).

Let R_n^- denote the worst-case minimal Bayes redundancy among all priors w,

$$R_n^- = \sup_w R_n(w). \tag{57}$$

This quantity also carries with it an information-theoretic interpretation. Here R_n^- is referred to as the *channel capacity*, $C(\Theta; X^n)$. Following Cover and Thomas (1991), we envision sending a message comprising a value of θ through a noisy channel represented by the conditional probability of X^n given θ. The receiver then attempts to reconstruct the message θ from X^n, or rather estimates θ from X^n. Assuming that θ is to be sampled from a distribution $w(\theta)$, the channel capacity represents the maximal message rate that the noisy channel allows. The capacity-achieving distribution "spaces" the input values of θ, countering the channel noise and aiding message recovery (see Cover and Thomas 1991).

Now observe that the channel capacity $C(\Theta; X^n)$ bounds the minimax redundancy R_n^+ (52) from below,

$$\begin{aligned} R_n^+ &= \min_q \sup_{\theta\in\Theta} R_n(f_\theta, q) \\ &\geq \sup_w \min_q \int_\Theta R_n(f_\theta, q) w(d\theta) \\ &= \sup_w \min_q R_n^*(q, w) \qquad (58) \\ &= \sup_w R_n(w) \qquad (59) \\ &\equiv C(\Theta; X^n), \end{aligned}$$

where (58) and (59) are simply the definitions of the Bayes redundancy (53) and the minimal Bayes redundancy (57).

Haussler (1997) demonstrated that in fact the minimax redundancy (52) is equal to the channel capacity,

$$R_n^+ = C(\Theta; X^n) = R_n^-. \tag{60}$$

According to this result, if we can calculate the capacity of the channel defined by the pair w and f_θ, then we can get the minimax redundancy immediately. This statement was first proved by Gallager (1976), although the minimax result of

this type for general loss functions was known before this point (Le Cam 1986). (See also Csiszár 1990; Davisson 1973; Davisson and Leon-Garcia 1980.)

To be useful, this equivalence requires us to compute the channel capacity for a pair w and f_θ. Unfortunately, this can be a daunting calculation. But when both the prior and density function are smooth, a familiar expansion can be used to derive a reasonable approximation. Let $I(\theta)$ denote the Fisher information matrix defined by

$$I_{i,j}(\theta) = E\left[\frac{\partial}{\partial\theta_i}\log f(X|\theta)\frac{\partial}{\partial\theta_j}\log f(X|\theta)\right] \quad \text{for all } i,j=1,\ldots,k.$$

Assume that the observation sequence $X^n = X_1,\ldots,X_n$ are iid (or *memoryless* in the parlance of information theory) from some distribution f_θ in the class $\mathbb{2}$. Under regularity conditions on the prior w and the model class $\mathbb{2}$, Clarke and Barron (1990) derived the following expansion in the general k-dimensional case (see Ibragimov and Has'minsky 1973, for the one-dimensional case). Let K be a compact subset in the interior of Θ. Then, given a positive, continuous prior density w supported on K, the expected redundancy (51) evaluated at the mixture distribution m^w (55) can be expanded as

$$R_n(f_\theta, m^w) = \frac{k}{2}\log\frac{n}{2\pi e} + \log\frac{\sqrt{\det I(\theta)}}{w(\theta)} + o(1),$$

where the $o(1)$ term is uniformly small on compact subsets interior to K. Averaging with respect to w yields an expansion for the minimal Bayes redundancy, or mutual information, (56),

$$\begin{aligned} R_n(w) &= I_w(\Theta; X^n) \\ &= \frac{k}{2}\log\frac{n}{2\pi e} + \int_K w(\theta)\log\frac{\sqrt{\det I(\theta)}}{w(\theta)}\,d\theta + o(1). \end{aligned}$$

The middle term is maximized by *Jeffreys's prior* (when this prior is well defined),

$$w^*(\theta) = \frac{\sqrt{\det I(\theta)}}{\int_K \sqrt{\det I(\theta)}\,d\theta},$$

Hence the minimax redundancy satisfies

$$\begin{aligned} R_n^+ &= \min_q \sup_{\theta\in\Theta} R_n(f_\theta, q) \\ &= \frac{k}{2}\log\frac{n}{2\pi e} + \log\int_K \sqrt{\det I(\theta)}\,d\theta + o(1). \end{aligned} \tag{61}$$

Recalling the equivalence (60) and the channel capacity interpretation of the worst-case minimal Bayes redundancy, Jeffreys's prior is now seen to be the *capacity-achieving* distribution for the channel defined by the pair w and $f_\theta(x^n)$. Intuitively, sampling a message θ according to Jeffreys's prior will result in channel inputs that are well separated in the sense that the probability of correctly reconstructing the message from X^n is high.

The leading term in (61) is the same $k/2\log n$ as in Rissanen's pointwise lower bound (51). Any code that achieves this leading term (to first order) on expected redundancy over a model class qualifies as a code to be used as the description length in the MDL selection for a model. (Barron et al. 1998 addressed qualifying coding schemes based on the constant term.) Such codes fairly represent all of the members in the model class (in the minimax sense) without the knowledge of exactly which distribution generated our data string.

To gain perspective, we now contrast the analysis of the Kullback–Leibler divergence $R_n(f_\theta, q)$ defined in (51) carried out for derivation of AIC with the analysis presented earlier. For AIC, we replace the distribution q with $f_{\hat\theta_n}$, where $\hat\theta_n$ is the MLE of θ. (Note that $f_{\hat\theta_n}$ is an estimator of the joint density of x^n, but is not a joint distribution. Thus it cannot be used to generate a code.) Under standard assumptions, the estimate $\hat\theta_n$ converges to θ in such a way that $R_n(f_\theta, f_{\hat\theta_n})$ has a negative $1/2\chi_k^2$ limiting distribution. Thus the Kullback–Liebler divergence $R_n(f_\theta, f_{\hat\theta_n})$ has a limiting mean of $-k/2$. This limit accounts for half of AIC's bias correction, the half associated with Kullback–Leibler divergence from f_θ due to parameter estimation (see Findley 1999, Sakamoto, Ishiguro, and Kitagawa 1985, p. 54). The minimax calculation in (61) is focused on a q, which is a joint density of x^n and determined by the set Θ. Moreover, Rissanen (1996) showed that the minimax redundancy is achieved asymptotically by the joint density (when it exists) corresponding to the NML code. That is, $f_{\hat\theta_n}(x^n)/C_n$, where C_n is the normalization constant required to make $f_{\hat\theta_n}(x^n)$ into a joint density or a code. The $-k/2$ term from the unnormalized MLE as in AIC case appears as $k/2\log 1/e$, and the rest of the terms in (61) give the asymptotic expansion of C_n (Barron et al. 1998). Hence MDL criteria that achieve minimax redundancy can be viewed as more conservative criteria than AIC from the perspective of Kullback–Leibler divergence.

For more general parameter spaces, Merhav and Feder (1995) proved that the capacity of the induced channel is a lower bound on the redundancy that holds simultaneously for all sources in the class except for a subset of points whose probability, under the capacity-achieving probability measure, vanishes as n tends to infinity. Because of the relationship between channel capacity and minimax redundancy, this means that the minimax redundancy is a lower bound on the redundancy for "most" choices of the parameter θ, hence generalizing Risssanen's lower bound.

For the case when the source is memoryless (i.e., when the observations are conditionally independent given the true parameter θ, and have a common distribution f_θ, $\theta\in\Theta$), Haussler and Opper (1997) obtained upper and lower bounds on the mutual information in terms of the relative entropy and Hellinger distance. Using these bounds and the relation between the minimax redundancy and channel capacity, one can obtain asymptotic values for minimax redundancy for abstract parameter spaces.

5.3 Achievability of Lower Bounds by Different Forms of Description Length

In regular parametric families (model classes), the forms of description length introduced in Section 3 all achieve the

$k/2\log n$ asymptotic lower bounds on redundancy in both the pointwise and minimax senses. They thus qualify as description lengths (to first order) to be used in MDL model selection. We illustrate this through our running Example 4 from Section 2.3. Our notation for a random data string now reverts to that from Section 4, so that x^n represents a random sequence $x_1, \ldots, x_n$.

Example 4 (Continued): Two-Stage Minimum Description Length. Trivially, because $\mathbb{2}_0$ consists of a single distribution, the expected redundancy of L_0 given in (4) is 0. Now, for $\theta \neq 0$

$$-\log f_\theta(x^n) = \frac{n}{2}\log(2\pi) + \frac{1}{2}\sum_{t=1}^{n}(x_t - \theta)^2.$$

Thus the expected redundancy between f_θ and the code length function L_1 (11) is given by

$$\begin{aligned} E_\theta\{\log f_\theta(x^n) - L_1(x^n)\} &= \frac{n}{2}E_\theta\{\bar{x}_n - \theta\}^2 + \frac{1}{2}\log n \\ &= \frac{1}{2} + \frac{1}{2}\log n, \end{aligned}$$

which for $k = 1$ achieves the pointwise lower bound (51).

Heuristically, for a general k-dimensional regular parametric family, it is well known that the quantity

$$-\log \frac{f_{\hat{\theta}}(x^n)}{f_\theta(x^n)}$$

has an asymptotic χ^2_k distribution; hence its expected value should be $k/2$, which is of smaller order than $k/2\log n$. Thus the two-stage description length achieves the lower bound.

Mixture Minimum Description Length. As with the two-stage scheme, the redundancy of L_0 is 0, because $\mathbb{2}_0$ consists of a single model. Now, starting with (15), we can calculate the expected redundancy for L_1,

$$\begin{aligned} &\frac{1}{2}\log(1+n\tau) + \frac{1}{2}\frac{n}{1+1/(n\tau)}E_\theta\bar{x}^2 - \sum_t \theta E_\theta x_t + \frac{1}{2}n\theta^2 \\ &\quad = \frac{1}{2}\log(1+n\tau) + \frac{1}{2}\frac{n}{1+1/(n\tau)}(1/n + \theta^2) - n\theta^2/2 \\ &\quad = \frac{1}{2}\log n + O(1), \end{aligned}$$

which clearly achieves the pointwise lower bound (51). In addition, given any prior distribution w on Θ, we can construct a prefix code according to the mixture distribution m^w (55). The corresponding code length is

$$L(x^n) = -\log \int w(d\theta) f_\theta(x^n).$$

As mentioned earlier, under certain regularity conditions, Clarke and Barron (1990) showed that the redundancy of the mixture code has the following asymptotic expansion for a regular family of dimension k:

$$R_n(m^w, \theta) = \frac{k}{2}\log\frac{n}{2\pi e} + \frac{1}{2}\log\frac{\sqrt{\det I(\theta)}}{w(\theta)} + o(1).$$

It follows that the mixture code achieves the minimax lower bound, and, as we mentioned earlier, Jeffreys's prior maximizes the constant term in the minimax redundancy (Barron et al. 1998).

Predictive Minimum Description Length. Using (22), it is easy to check the redundancy

$$\begin{aligned} E_\theta(-\log q(x^n) + \log f_\theta(x^n)) &= \frac{1}{2}\sum_{t=1}^{n}(1 + 1/t) - n/2 \\ &= \frac{1}{2}\sum_{t=1}^{n} 1/t \\ &= \frac{1}{2}\log n + O(1). \end{aligned}$$

Thus it achieves the lower bound (51) and can be used as the description length for data based on model $\mathbb{2}_1$. As with the previous two forms, the expected redundancy of L_0 is 0.

For more general cases, Rissanen (1986b, thm. 3) proved that the predictive code based on the MLE achieves the pointwise redundancy lower bound under regularity conditions.

5.4 Assessing Minimum Description Length Model Selection Procedures in Terms of Consistency and Prediction Errors

Although MDL has a solid motivation from the viewpoint of noiseless compression of data, which itself has a close tie to statistical estimation, it is not clear a priori whether or not MDL will lead to model selection procedures that are sensible statistically. One criterion used in assessing model selection procedures is consistency when a finite-dimensional "true" model is assumed. That is, as the sample size gets large, a consistent procedure will pick the correct model class with probability approaching 1. The two-stage, predictive, and mixture forms of MDL are consistent in the regression case (Speed and Yu 1994). In general, different MDL forms are consistent under very weak conditions (Barron et al. 1998). The predictive code takes the form of predictive least squares in time series and stochastic regression models. (See Hemerly and Davis 1989 for time series models and Wei 1992 for general stochastic regression models and the consistency of the predictive form.) We illustrate the consistency of MDL through the two-stage code in our running example. Under the same finite-dimensional "true" model assumption, as an alternative to the consistency assessment, Merhav (1989) and Merhav, Gutman, and Ziv (1989) analyzed model selection criteria by studying the best possible underfitting probability while exponentially restricting the overfitting probability.

Example 4 (Continued). Recall that two-stage MDL or BIC will select $\mathbb{2}_0$ if $|\bar{x}_n| \leq \sqrt{\log n/n}$. When $\mathbb{2}_1$ is true, the probability of underfitting is

$$\begin{aligned} \Pr(\mathbb{2}_0 \text{ is selected}) &= P_\theta(|\bar{x}_n| \leq \sqrt{\log n/n}) \\ &\approx P_\theta(N(0,1) \geq \theta\sqrt{n} - \sqrt{\log n}) \\ &\approx O(e^{-n\theta^2/2}). \end{aligned}$$

Similarly, when $\mathbb{2}_0$ is true, the probability of overfitting is

$$\begin{aligned}\Pr(\mathbb{2}_1 \text{ is selected}) &= P_\theta(|\bar{x}_n| > \sqrt{\log n/n}) \\ &= P_\theta(|N(0,1)| > \sqrt{\log n}) \\ &\approx O(1/\sqrt{n}).\end{aligned}$$

Thus two-stage MDL yields a consistent model selection rule.

In general, an exponential decay rate on the underfitting probability and an algebraic decay rate on the overfitting probability hold for the predictive and mixture MDL forms, and also for other regression models (Speed and Yu 1994). Consistency of MDL follows immediately. It also follows from an examination of the underfitting probability that for finite sample sizes, consistency is effected by the magnitude of θ^2 (or squared bias in general) relative to n, and not by the absolute magnitude of θ^2. Speed and Yu (1994) also studied the behavior of MDL criteria in two prediction frameworks: prediction without refitting and prediction with refitting. In both cases, MDL (and BIC) turned out to be optimal if the true regression model is finite-dimensional. AIC is not consistent, but the consequence in terms of prediction errors is not severe; the ratio of AIC's prediction error and that of any form of MDL (or BIC) is bounded.

No model is true in practice, but the finite-dimensional model assumption in regression does approximate the practical situation in which the model bias has a "cliff" or a sharp drop at a certain submodel class under consideration, or when the covariates can be divided into two groups of which one is very important and the other marginal and no important covariates are missing from consideration. But when bias decays gradually and never hits 0, the consistency criterion does not make sense. In this case prediction error provides insight into the performance of a selection rule. Shibata (1981) showed that AIC is optimal for these situations, at least in terms of one-step-ahead prediction error. The simulation studies in Section 4 illustrate that by trading off between bias and variance, it is possible to create examples in which BIC outperforms AIC and vice versa. A similar point was made by Speed and Yu (1994). When the covariates under consideration are misspecified or superfluous, Findley (1991) gave examples both in regression and time series models in which the bigger model always gives a smaller prediction error thus suggesting that AIC is better for these particular models. For exactly these reasons, we believe that adaptive model selection criteria likc gMDL are very useful.

6. CONCLUSIONS

In this article we have reviewed the principle of MDL and its various applications to statistical model selection. Through a number of simple examples, we have motivated the notion of code length as a measure for evaluating competing descriptions of data. This brings a rich information-theoretic interpretation to statistical modeling. Throughout this discussion, our emphasis has been on the practical aspects of MDL. Toward that end, we developed in some detail MDL variable selection criteria for regression, perhaps the most widely applied modeling framework. As we have seen, the resulting procedures have connections to both frequentist and Bayesian methods. Two mixture forms of MDL, iMDL and gMDL, exhibit a certain degree of adaptability, allowing them to perform like AIC at one extreme and BIC at the other. To illustrate the scope of the MDL framework, we have also discussed model selection in the context of curve estimation, cluster analysis, and order selection in ARMA models.

Some care has gone into the treatment of so-called valid description lengths. This notion is important, as it justifies the use of a given coding scheme for comparing competing models. Any implementation of MDL depends on the establishment of a universal coding theorem, guaranteeing that the resulting selection rule has good theoretical properties, at least asymptotically. The two-stage, mixture, predictive, and normalized ML coding schemes all produce valid description lengths. Our understanding of the finite-sample performance of even these existing MDL criteria, will improve as they find greater application within the statistics community. To aid this endeavor, the MDL procedures discussed in this article will be made available by the first author in the form of an S-PLUS library.

Inspired by algorithmic complexity theory, the descriptive modeling philosophy of MDL adds to other, more traditional views of statistics. Within engineering, MDL is being applied to ever-more exotic modeling situations, and there is no doubt that new forms of description length will continue to appear. MDL provides an objective umbrella under which rather disparate approaches to statistical modeling can coexist and be compared. In crafting this discussion, we have tried to point out interesting open problems and areas needing statistical attention. At the top of this list is the incorporation of uncertainty measures into the MDL framework. The close ties with Bayesian statistics yields a number of natural suggestions in this direction, but nothing formal has been done in this regard. The practical application of MDL in nonparametric problems should also provide a rich area of research, because theoretical results in this direction are already quite promising (see, e.g., Barron and Yang, 1998; Yang 1999).

APPENDIX: TECHNICAL DETAILS FOR MIXTURE MDL

We begin with the normal inverse-gamma family of conjugate priors for the normal linear regression model (23). Setting $\tau = \sigma^2$, these densities are given by

$$w(\beta,\tau) \propto \tau^{\frac{-d+k+2}{2}} \exp\left[\frac{-(\beta-b)^t V^{-1}(\beta-b)+a}{2\tau}\right] \qquad \text{(A.1)}$$

and depend on several hyperparameters: $a, d \in \mathbb{R}$, the vector $b \in \mathbb{R}^k$, and a $k \times k$ symmetric, positive definite matrix V. Valid ranges for these parameters include all values that make (A.1) a proper density. Under this class of priors, the mixture distribution (30) has the form

$$-\log m(y|X) = \frac{1}{2}\log|V| - \frac{1}{2}\log|V^*| - \frac{d}{2}\log a + \frac{d^*}{2}\log a^*, \qquad \text{(A.2)}$$

ignoring terms that do not depend on our particular choice of model, where

$$d^* = d+n, \qquad V^* = (V^{-1}+X^tX)^{-1}, \qquad b^* = V^*(V^{-1}b+X^ty),$$

and

$$a^* = a + y^ty + b^tV^{-1}b - (b^*)^t(V^*)^{-1}b^*.$$

The derivation of $m(y|X)$, the marginal or predictive distribution of y, is standard and was given by O'Hagan (1994).

To implement this mixture form of MDL, we must settle on values for the hyperparameters. In his original derivation, Rissanen (1989) considered normal inverse-gamma priors with

$$d=1, \qquad V=c^{-1}\Sigma, \qquad \text{and} \quad b=(0,\ldots,0). \tag{A.3}$$

After making these substitutions, we then want to minimize the expression (A.2) over the two hyperparameters a and c. First, a straightforward calculation gives us the closed-form expression $\hat{a}=R_c/n$. Substituting $\hat{a}$ for a, we arrive at the log-likelihood

$$-\log m(y|X,\hat{a},c)=-\frac{1}{2}\log\left|c\Sigma^{-1}\right|+\frac{1}{2}\log\left|c\Sigma^{-1}+X^tX\right|+\frac{n}{2}\log R_c. \tag{A.4}$$

Surprisingly, we obtain this form no matter how we select d in our prior specification (A.3), so $d=1$ is not a restrictive choice. This form is in fact equivalent to a mixture distribution computed under the so-called weak prior corresponding to $a=d=0$, a choice of hyperparameters that assigns the improper prior $1/\tau$ to τ.

Unfortunately, optimizing over c presents us with a more difficult problem. After differentiating (31), we find that $\hat{c}$ must satisfy

$$\hat{c}=\frac{kR_{\hat{c}}}{R_{\hat{c}}\operatorname{trace}[\Sigma^{-1}(\hat{c}\Sigma^{-1}+X^tX)^{-1}]+ny^tX(\hat{c}\Sigma^{-1}+X^tX)^{-1}\Sigma^{-1}(\hat{c}\Sigma^{-1}+X^tX)^{-1}X^ty}. \tag{A.5}$$

This expression can be be applied iteratively, with convergence typically requiring fewer than 20 steps, depending on the starting values. In deriving what we have called iMDL, Rissanen (1989, p. 129) exhibited a slightly different relationship for the special case of $\Sigma=I_{k\times k}$. (The difference is presumably the result of transcription errors.) To obtain gMDL, we instead choose $\Sigma=(X^tX)^{-1}$, and we arrive at the expression for $\hat{c}$ given in (33) either by direct substitution in (A.5) or by minimizing (A.4).

[Received October 1999. Revised December 1999.]

REFERENCES

Akaike, H. (1974), "A New Look at the Statistical Model Identification," *IEEE Transactions on Automatic Control*, 19 716–723.

——— (1977), "An Objective Use of Bayesian Models," *Annals of the Institute of Statistical Mathematics*, 29, part A, 9–20.

An, H., and Gu, L. (1985), "On the Selection of Regression Variables," *Acta Mathematica Applicata Sinica*, 2, 27–36.

Barron, A., Rissanen, J., and Yu, B. (1998), "The Minimum Description Length Principle in Coding and Modeling," *IEEE Transactions on Information Theory*, 44, 2743–2760.

Baxter, R., and Oliver, J. (1995), "MDL and MML: Similarities and Differences" (introduction to Minimum Encoding Inference—Part III), unpublished manuscript.

Berger, J., and Pericchi, L. (1996), "The Intrinsic Bayes Factor for Model Selection and Prediction," *Journal of the Amererican Statistical Association*, 91, 109–122.

Bernardo, J. M., and Smith, A. F. M. (1994), *Bayesian Theory*, New York: Wiley.

Brockwell, P. J., and Davis, R. A. (1991), *Time Series: Theory and Methods*, New York: Springer-Verlag.

Broman, K. W. (1997), "Identifying quantitative trait loci in experimental crosses," unpublished doctoral dissertation, University of California, Berkeley, Dept. of Statistics.

Carlin, B. P., Polson, N. G., and Stoffer, D. S. (1992), "Monte Carlo Approach to Nonnormal and Nonlinear State-Space Modeling." *Journal of the American the Statistical Association*, 87 493–500.

Clarke, B. S., and Barron, A. R. (1990). "Information-Theoretic Asymptotics of Bayes Methods," *IEEE Transactions on Information Theory*, 36, 453–471.

Clyde, M., DeSimone, H., and Parmigiani, G. (1996), "Prediction via Orthogonalized Model Mixing," *Journal of the American Statistical Association*, 91, 1197–1208.

Clyde, M., Parmigiani, G., and Vidakovic, B. (1998), "Multiple Shrinkage and Subset Selection in Wavelets," *Biometrika*, 85, 391–402.

Cover, T. M., and Thomas, J. A. (1991), *Elements of Information Theory*, New York: Wiley.

Cowen, N. M. (1989), "Multiple Linear Regression Analysis of RFLP Data Sets Used in Mapping QTLs," in *Development and Application of Molecular Markers to Problems in Plant Genetics*, eds. T. Helentjaris and B. Burr, Cold Spring Harbor, NY: Cold Spring Harbor Laboratory, pp. 113–116.

Csiszár, I. (1990), "Information Theoretical Methods in Statistics," class notes, University of Maryland, College Park, MD.

Davisson, L. (1973), "Universal Noiseless Coding," *IEEE Transactions on Information Theory*, 19, 783–795.

Davisson, L., and Leon-Garcia, A. (1980), "A Source Matching Approach to Finding Minimax Codes," *IEEE Transactions on Information Theory*, 26, 166–174.

Dawid, A. P. (1984), "Present Position and Potential Developments: Some Personal Views, Statistical Theory, the Prequential Approach," *Journal of the Royal Statistical Society*, Series B, 147, 178–292.

——— (1991), "Prequential Analysis, Stochastic Complexity and Bayesian Inference," prsented at the Fourth Valencia International Meeting on Bayesian Statistics, Peniscola, Spain.

Doerge, R. W., and Churchill, G. A. (1996). "Permutation Tests for Multiple Loci Affecting a Quantitative Character," *Genetics*, 134, 585–596.

Edwards, A. W. F. (1972), *Likelihood*, Cambridge, U.K.: Cambridge University Press.

Elias, P. (1975), "Universal Code Length Sets and Representations of Integers," *IEEE Transactions on Information Theory*, 21, 194–203.

Findley, D. F. (1991), "Counterexamples to Parsimony and BIC," *Annals of Statistics*, 43, 505–514.

——— (1999), "AIC II," in *Encyclopedia of Statistical Sciences, Update Vol. 3.* (eds. S. Kotz, C. R. Read, and D. L. Banks), New York, Wiley.

Furby, S., and Kiiveri, H., and Campbell, N. (1990), "The Analysis of High-Dimensional Curves," in *Proceedings of the 5th Australian Remote Sensing Conference*, pp. 175–184.

Furnival, G., and Wilson, R. (1974), "Regressions by Leaps and Bounds," *Technometrics*, 16, 499–511.

Gallager, R. G. (1976), "Source Coding with Side Information and Universal Coding," unpublished manuscript.

Gerencsér, L. (1987), "Order Estimation of Stationary Gaussian ARMR Processes Using Rissanen's Complexity," working paper, Computer and Automation Institute of the Hungarian Academy of Sciences.

——— (1994), "On Rissanen's Predictive Stochastic Complexity for Stationary ARMA Processes," *Journal of Statistical Planning and Inference*, 41, 303–325.

George, E., and Foster, D. (1999), "Calibration and Empirical Bayes Variable Selection," *Biometrika*, in press.

George, E., and McCulloch, R. E. (1993), "Variable Selection via Gibbs Sampling," *Journal of the American Statistical Association*, 88, 881–889.

——— (1997), "Approaches for Bayesian Variable Selection," *Statistica Sinica*, 7, 339–373.

Hannan, E. J., and Kavalieris, L. (1984), "A Method for Autoregressive-Moving Average Estimation," *Biometrika*, 71, 273–280.

Hannan, E. J., McDougall, A. J., and Poskitt, D. S. (1989), "Recursive Estimation of Autoregressions," *Journal of the Royal Statistical Society*, Series B, 51, 217–233.

Hannan, E. J., and Quinn, B. G. (1979), "The Determination of the Order of an Autoregression," *Journal of the Royal Statistical Society*, Series B, 41, 190–195.

Hannan, E. J., and Rissanen, J. (1982), "Recursive Estimation of Mixed Autoregressive-Moving Average Order," *Biometrika*, 69, 81–94.

Hansen, M., and Kooperberg, C. (1999), "Spline Adaptation in Extended Linear Models," submitted to *Statistical Science*.

Hansen, M., and Yu, B. (1999), "Bridging AIC and BIC: An MDL Model Selection Criterion," in *Proceedings of the IT Workshop on Detection, Estimation, Classification and Imaging*, Santa Fe, NM.

Hastie, T., Buja, A., and Tibshirani, R. (1995), "Penalized Discriminant Analysis." *The Annals of Statistics*, 23, 73–102.

Hastie, T., Tibshirani, R., and Buja, A. (1994), "Flexibile Discriminant Analysis by Optimal Scoring," *Journal of the American Statistical Association*, 89, 1255–1270.

Haussler, D. (1997), "A General Minimax Result for Relative Entropy," *IEEE Transactions on Information Theory*, 43, 1276–1280.

Haussler, D., Kearns, M., and Schapire, R. E. (1994), "Bounds on the Sample Complexity of Bayesian Learning Using Information Theory and the VC dimension," *Machine Learning*, 14, 83–113.

Haussler, D., and Opper, M. (1997), "Mutual Information, Metric Entropy, and Risk in Estimation of Probability Distributions," *The Annals of Statistics*, 25, 2451–2492.

Hemerly, E. M., and Davis, M. H. A. (1989), "Strong Consistency of the Predictive Least Squares Criterion for Order Determination of Autoregressive

Processes." *The Annals of Statistics*, 17 941–946.
Hertz, J., Krough, A., and Palmer, R. G. (1991), *Introduction to the Theory of Neural Computation*, Redwood City, CA: Addison-Wesley.
Huang, D. (1990), "Selecting Order for General Autoregressive Models by Minimum Description Length," *Journal of Time Series Analysis*, 11, 107–119.
Hurvich, C. M., Simonoff, J. S., and Tsai, C.-L. (1998), "Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion," *Journal of the Royal Statistical Society*, Series B, 60, 271–293.
Hurvich, C. M., and Tsai, C.-L. (1989), "Regression and Time Series Model Selection in Small Samples," *Biometrika*, 76, 297–307.
Ibragimov, I. A., and Has'minsky, R. Z. (1973), "On the Information in a Sample About a Parameter," in *Proceedings of the 2nd International Symposium on Information Theory*. Eds., B. N. Petrov and F. Csáki, Akademiai Kiado, Budapest.
Jobson, J. D. (1992), *Applied Multivariate Data Analysis, Vol. II: Categorical and Multivariate Methods*, New York: Springer-Verlag.
Kass, R. E., and Raftery, A. E. (1995), "Bayes Factors," *Journal of the American Statistical Association*, 90, 773–795.
Kolmogorov, A. N. (1965), "Three Approaches to the Quantitative Definition of Information," *Problems Information Transmission*, 1, 1–7.
——— (1968), "Logical Basis for Information Theory and Probability Theory, *IEEE Transactions on Information Theory*, 14, 662–664.
Le Cam, L. M. (1986), *Asymptotic Methods in Statistical Decision Theory*. New York: Springer-Verlag.
Leclerc, Y. G. (1989), "Constructing Simple Stable Descriptions for Image Partitioning," *International Journal of Computer Vision*, 3, 73–102.
Lai, T. L., and Lee, C. P. (1997), "Information and Prediction Criteria for Model Selection in Stochastic Regression and ARMA Models," *Statistica Sinica*, 7, 285–309.
Li, M., and Vatányi, P. (1996), *An Introduction to Kolmogorov Complexity and Its Applications*, New York: Springer-Verlag.
Lindley, D. V. (1956), "On a Measure of the Information Provided by an Experiment," *Annals of Mathematical Statistics*, 27, 986–1005.
Long, A. D., Mullaney, S. L., Reid, L. A., Fry, J. D., Langley, C. H., and Mackay, T. F. C. (1995), "High-Resolution Mapping of Genetic Factors Affecting Abdominal Bristle Number," in *Drosophila Melanogaster. Genetics*, 139, 1273–1291.
Luo, Z., and Wahba, G. (1997), "Hybrid Adaptive Splines," *Journal of the American Statistical Association*, 92, 107–116.
Madigan, D., Raftery, A., and Hoeting, J. (1997), "Bayesian Model Averaging for Linear Regression Models," *Journal of the American Statistical Association*, 92, 179–191.
Mallows, C. L. (1973), "Some Comments on C_p," *Technometrics*, 15, 661–675.
——— (1995), "More Comments on C_p," *Technometrics*, 37, 362–372.
Merhav, N. (1989), "The Estimation of the Model Order in Exponential Families," *IEEE Transactions on Information Theory*, 35, 1109–1114.
Merhav, N., and Feder, M. (1995), "A Strong Version of the Redundancy-Capacity Theorem of Universal Coding, *IEEE Transactions on Information Theory*, 41, 714–722.
Merhav, N., Gutman, M, and Ziv, J. (1989), "On the Estimation of the Order of a Markov Chain and Universal Data Compression," *IEEE Transactions on Information Theory*, 35, 1014–1019.
Moulin, P. (1996), "Signal Estimation Using Adapted Tree-Structured Bases and the MDL Principle," *Proceedings of the Time-Frequency and Time-Scale Analysis*, pp. 141–143.
O'Hagan, A. (1994), *Kendall's Advanced Theory of Statistics: Bayesian Inference, Vol. 2B*, New York: Wiley.
——— (1995), "Fractional Bayes Factors for Model Comparison, *Journal of the Royal Statistical Society*, Series B, 57, 99–138.
Pan, H.-P., and Forstner, W. (1994), "Segmentation of Remotely Sensed Images by MDL-Principled Polygon Map Grammar," *Int. Archives of Photogrammetry and Remote Sensing*, 30, 648–655.
Peterson, J. J. (1986), "A Note on Some Model Selection Criteria" *Statistics and Probability Letters*, 4, 227–230.
Qian, G., Gabor, G., and Gupta, R. P. (1996), "Generalised Linear Model Selection by the Predictive Least Quasi-Deviance Criterion," *Biometrika*, 83, 41–54.
Rissanen, J. (1978), "Modeling by Shortest Data Description," *Automatica*, 14, 465–471.
——— (1983), "A Universal Prior for Integers and Estimation by Minimum Description Length," *The Annals of Statistics*, 11, 416–431.
——— (1986a), "Stochastic Complexity and Modeling," *The Annals of Statistics*, 14, 1080–1100.
——— (1986b), "A Predictive Least Squares Principle," *IMA Journal of Mathematical Control and Information*, 3, 211–222.
——— (1987), "Stochastic Complexity" (with discussions), *Journal of the Royal Statistical Society*, Series B, 49, 223–265.
——— (1989), *Stochastic Complexity and Statistical Inquiry*, Singapore: World Scientific.
——— (1996), "Fisher Information and Stochastic Complexity," *IEEE Transactions on Information Theory*, 42, 48–54.
Rissanen, J., Speed, T. P., and Yu, B. (1992), "Density Estimation by Stochastic Complexity," *IEEE Transactions on Information Theory*, 38, 315–323.
Sakamoto, Y., Ishiguro, M,, and Kitagawa, G. (1985), *Akaike Information Statistics*. Dordrecht: Reidel.
Saito, N. (1994), "Simultaneous Noise Suppression and Signal Compression Using a Library of Orthonormal Bases and the Minimum Description Length Criterion," in *Wavelets in Geophysics*, eds. E. Foufoula-Georgiou and P. Kumar, New York: Academic Press, pp. 299–324.
Schumaker, L. L. (1993), *Spline Functions: Basic Theory*, Malabar, FL: Krieger.
Schwarz, G. (1978), "Estimating the Dimension of a Model," *The Annals of Statistics*, 6, 461–464.
Sclove, S. L. (1968), "Improved Estimators for Coefficients in Linear Regression," *Journal of the American Statistical Association*, 63 596–606.
Sclove, S. L., Morris, C., and Radhakrishnan, R. (1972), "Non-Optimality of Preliminary-Test Estimators for the Mean of a Multivariate Normal Distribution," *Annals of Mathematical Statistics*, 43, 1481–1490.
Shibata, R. (1981), "An Optimal Selection of Regression Variables," *Biometrika*, 68, 45–54.
Shtarkov, Y. M. (1987), "Universal Sequential Coding of Single Messages," *Problems of Information Transmission*, 23, 3–17.
Smith, M. (1996), "Nonparametric Regression: A Markov Chain Monte Carlo Approach," doctoral thesis, Australian Graduate School of Management University of New South Wales, Australia.
Smith, M., and Kohn, R. (1996), "Nonparametric Regression Using Bayesian Variable Selection," *Journal of Econometrics*, 75, 317–344.
Smith, A. F. M., and Spielgelhalter, D. J. (1980), "Bayes Factors and Choice Criteria for Linear Models," *Journal of the Royal Statistical Society* Series B, 42, 213–220.
SSDC, Social Sciences Data Collection at University of California, San Diego http: //ssdc.ucsd.edu/ssdc/NYSE.Date.Day.Return.Volume.Vola.text.
Speed, T. P., and Yu, B. (1994), "Model Selection and Prediction: Normal Regression," *Annals of the Institute of Statistical Mathematics*, 45 35–54.
Stine, R. A., and Foster, D. P. (1999), "The Competitive Complexity Ratio," unpublished manuscript.
Sugiura, N. (1978), "Further Analysis of the Data by Akaike's Information Criterion and the Finite Corrections," *Communications in Statistics*, A7, 13–26.
Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society* Series B, 58, 267–288.
Wahba, G. (1990), *Spline Models for Observational Data*, Philadelphia: SIAM.
Wallace, C. S., and Boulton, D. M. (1968), "An Information Measure for Classification," *Computing Journal*, 11, 185–195.
Wallace, C. S., and Dowe, D. L. (1994), "Intrinsic Classification by MML—the SNOB Program," *Proceedings of 7th Australian Joint Conference on Artificial Intelligence*, UNE, Armidale, NSW, World Scientific, Singapore, pp 37–44.
Wallace, C. S., and Freeman, P. R. (1987), "Estimation and Inference by Compact Coding (with discussion)," *Journal of the Royal Statistical Society*, Series B, 49, 240–251.
Wei, C. Z. (1992), "On the Predictive Least Squares Principle," *The Annals of Statistics*, 36, 581–588.
Wong, F., Hansen, M., Kohn, R., and Smith, M. (1997), "Focused Sampling and Its Application to Non-Parametric and Robust Regression," Technical Report, Bell Laboratories.
Yu, B., and Speed, T. P. (1992), "Data Compression and Histograms," *Probability Theory and Related Fields*, 92, 195–229.
Zellner, A. (1986), "On Assessing Prior Distributions and Bayesian Regression Analysis with g-Prior Distributions," in *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*, eds. P. K. Goel and A. Zellner, Amsterdam: North-Holland, 233–243.