Model Selection and the Principle of Minimum Description Length

Mark H. Hansen and Bin Yu

Abstract

This paper 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 interest within the statistics community. In the pages that follow, we review both the practical as well as 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.

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 throughout this paper is on the practical application of MDL, and hence we make extensive use of real data sets.

KEY WORDS: AIC; Bayesian Methods; BIC; Cluster Analysis; Code Length; Coding Redundancy; Information Theory; Minimum Description Length; Model Selection; Pointwise and Minimax Lower Bounds; Regression; Time Series.

1Mark H. Hansen is Member of the Technical Staff, Bell Laboratories, Murray Hill, NJ 07974. Bin Yu is Associate Professor, Department of Statistics, University of California, Berkeley, CA; and Member of the Technical Staff, Bell Laboratories, Murray Hill, NJ 07974. Bin Yu would like to acknowledge support from Grant DMS-9322817 from the National Science Foundation and Grant DAAH04-94-G-0232 from the Army Research Office.
1. Overview

The Principle of Parsimony or Occam’s Razor is implicitly embedded in all sensible statistical modeling and is the soul of model selection. Model selection is encountered when investigators have to 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 scientific theory. To implement the Parsimony Principle in a formal way, one has to quantify “parsimony” of a model relative to the available data. One wants to find a concise model that captures or fits the important features of the data. A concise model should be easy to describe, while a good fit implies that the model makes easy the description of the data. Rissanen (1978) distills such thinking in his Principle of Minimum Description Length (MDL):

Choose the model that gives the shortest description of data.

MDL has its intellectual roots in the algorithmic or descriptive complexity theory of Kolmogorov, Chaitin and Solomonoff (Li and Vitányi, 1996). It is interesting that Kolmogorov, the founder of axiomatic probability theory, would in later life turn to this alternative framework. This new concept of probability is based on the length of the shortest binary computer program that describes an object (or event), which we refer to as the descriptive complexity of the object. (One can envision such a program printing or in some way exhibiting the object.) Amazingly, descriptive complexity can be defined independent of any specific computing device (Kolmogorov, 1965, 1968; and Cover and Thomas, 1991), making it a universal quantity. When an object is a random entity $X$, then it is intuitively clear that, for efficiency, frequently appearing values of $X$ should be assigned short programs or descriptions, while rare values are given longer programs. Hence the length of the description corresponds to the probability of an event in the more traditional sense. 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 Kolmogorov complexity is not computable (cf. Cover and Thomas, 1991). Rissanen’s MDL follows the same train of thought, but avoids the non-computability problem by restricting the search to only those descriptions (programs) that correspond to probability distributions; and for any distribution $Q$ the description length is known, via Kraft’s inequality, to be $-\log_2 Q(data)$.

An important precursor to Rissanen’s MDL is Wallace and Boulton (1968), which applies the idea of Minimum Message Length (MML) or description length to clustering problems. This work, however, stops short of a general framework for other statistical problems. (See Wallance and Freeman, 1987; and Baxter and Oliver, 1995, for recent developments on MML.) Two other papers, which are influential and important in their own right, are Akaike (1974) and Schwarz (1978). In his derivation of $AIC$, Akaike’s Information Criterion, Akaike (1974) gives for the first time formal recipes for general model selection problems from the point of view of prediction. Schwarz (1978) takes a Bayesian approach to model selection deriving an approximation to a Bayesian posterior. This approximate Bayesian model selection criterion has a form very similar to $AIC$ and is termed the Bayesian Information Criterion, $BIC$. It is fascinating to note the crucial role that the information-theoretic Kullback-Leibler divergence played in the derivation of $AIC$, since we will see in this article that Kullback-Leibler divergence is indispensable in the MDL paradigm. As many other statistical procedures, MDL and $AIC$ were brought about by non-statisticians.
In the MDL-paradigm, any probability distribution is considered from a descriptive point of view, that is, it is not necessarily the underlying data-generating mechanism (although it does not exclude such a possibility). Thus the MDL paradigm 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 the celebrated Shannon’s source coding theorem (cf. Cover and Thomas, 1991) states that this “true” distribution gives the minimum description length of the data (on average and asymptotically).

MDL connects 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, MDL gives rise to the well-accepted Maximum Likelihood (ML) Principle in frequentist statistics. It is therefore a generalization of the Maximum Likelihood 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. 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.

The rest of the paper is organized as follows. Section 2 introduces basic coding concepts and explains the MDL principle. In particular, we start with Kraft’s inequality, which establishes the equivalence between probability distributions and description length. We illustrate different coding ideas through a simple example of coding or compressing up-down indicators derived from daily statistics of the Dow-Jones Industrial Average. We emphasize that using a probability distribution for description or coding purposes does not require that it actually generates our data. We revisit MDL in the end of Section 2 to connect it with the Maximum Likelihood Principle and Bayesian statistics, and we bring up the issue of valid description length before any implementation of MDL. (This issue is explored in depth in Section 5.) Section 3 formally introduces different forms of MDL such as two-stage (or multi-stage in general), mixture, predictive, and normalized maximized likelihood.

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. A new model selection criterion based on MDL, gMDL, is proposed. We compare this new model criterion, other MDL criteria, BIC, and AIC through both 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, 1998, further explores 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 ARMA model to the Dow-Jones data sets, comparing predictive MDL (PMDL), BIC and AIC for order selection.

Section 5 reviews theoretical results on MDL. They 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 (1986) 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 MDL Principle

2.1 Probability and idealized code length

2.1.1 The discrete case

A code $C$ on a set $A$ is simply a mapping from $A$ to a set of codewords. In this section, we will consider binary codes so that each codeword is a string of 0’s and 1’s. Let $A$ be a finite set and let $Q$ denote a probability distribution on $A$. 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 $A$.

Example 1 (Huffman’s Algorithm) Let $A = \{a, b, c\}$ and let $Q$ denote a probability distribution on $A$ 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 $A$ 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 $bc$ having node probability $1/4 + 1/4 = 1/2$. We then iterate the process with the new set of nodes $\{a, bc\}$. Since 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 as well as 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 $x \in A$.

Clearly, the Huffman code constructed in our example is not unique because we can permute the labels at each level in the tree. In addition, 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. 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 $aaaacb$. 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 referred to as a \textit{prefix code}. By taking their codewords as endnodes 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 satisfies Kraft’s inequality

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

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

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

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 $Q$.\footnote{We can only guarantee that the length function $L$ derived from Huffman’s algorithm is within two bits of $\lceil -\log_2 Q \rceil$. While 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 to a known distribution $P$, or in statistical terms, we observe data drawn from $P$. Given a code $C$ on $\mathcal{A}$ with length function $L$, the \textit{expected code length} of $C$ with respect to $P$ is defined to be

$$L_C = \sum_{x \in \mathcal{A}} P(x)L(x).$$

As we have seen, if $C$ is a prefix code, $L$ 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 elements of $\mathcal{A}$ are generated according to a probability distribution $P$. For any prefix code $C$ on $\mathcal{A}$ with length function $L$, the expected code length $L_C$ is bounded below by $H(P)$, the entropy of $P$. That is,

$$L_C \geq H(P) \equiv -\sum_{a \in \mathcal{A}} P(a) \log P(a),$$

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. 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 \( A \), not necessarily the data-generating or true distribution \( P \), can be used to encode data from \( A \). In most statistical applications, the true distribution \( P \) is rarely known, and to a large extent, this paper 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 will 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 binary digit. (Later in the paper, we will also use the unit nat when a natural log is taken.)

**Example 2 (Code length for finitely many integers)** Consider the set of integers \( A = \{1, 2, 3, \ldots, N\} \) and let \( Q \) denote the uniform distribution on \( A \), so that \( Q(k) = 1/N \) for all \( k \in A \). By applying Huffman’s algorithm in this setting, we obtain a uniform code with length function that is not greater than \( \log_2 N \) for all \( k \), but is equal to \( \log_2 N \) for at least two values of \( k \). While we know from Shannon’s Source Coding Theorem that an expected code length of \( \log_2 N \) 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.

For the code length of an “optimal code” on natural numbers, see Rissanen (1983).

### 2.1.2 The continuous case

Suppose that our data is no longer restricted to a finite set, but instead ranges over an arbitrary subset of the real line. Let \( f \) denote the data-generating or true density. Given another density \( q \) defined on \( A \), we can construct a code for our data by first discretizing \( A \) and then applying, say, Huffman’s algorithm. In most statistical applications, we are not interested in \( A \), but rather its Cartesian product \( A^n \) corresponding to an \( n \)-dimensional continuous data sequence \( x^n = (x_1, \ldots, x_n) \). Then, if we discretize \( A \) into equal cells of size \( \delta \), 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 \( \delta \) is the precision of the discretization, and for fixed \( \delta \) we refer to \(-\log_2 q(x^n)\) as an idealized code length. In Section 3.1, we will 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 description of the data is given by a code based on the true density of \( f(x^n) \) of data sequence \( x^n \). In this case, the lower bound on the expected code length is simply the differential entropy

\[
H(f) = -\int \log_2 f(x^n) f(x^n) dx^n. \tag{4}
\]
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 we 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 will be 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 will be discussed at length in the next section when we take up modeling issues in greater detail.

Example 3 (Code length for finite, binary strings) Consider two “up and down” indicators derived from the daily return and intra-day volatility of the Dow-Jones Industrial Average. To each of the 6430 trading days between July, 1962 and June, 1988, we assigned a pair of binary variables. The first takes the value 1 if the return on a given day was higher than that for the previous day (an “up”), and 0 otherwise (a “down”). The second variable is defined similarly, but instead tracks the volatility series. This gives us two binary strings of length $n = 6430 - 1 = 6429$. There are 3181 or 49.49% 1’s or up’s in the return string, compared with 2023 or 31.47% 1’s in the volatility string. In Figure 2, we present the last 1,000 observations from each series. To coordinate with our construction of binary strings, we have plotted daily differences so that up’s correspond to positive values and down’s to negative values. In the panels below these plots, we identify days on which each series increased with a black vertical bar. 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 intra-day volatility jumped dramatically, while the overall return was down sharply from the previous day.

Using these strings, we will describe three coding algorithms, each assuming that the length of
the string, \( n = 6429 \), 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 to its branch in San Diego. Clearly, each string can be transmitted directly without any further coding, requiring \( n = 6429 \) 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 the sender employs the uniform coding scheme of Example 2 with \( N = n = 6429 \) to encode and send the total number of \( k \) up’s in either the return or the volatility series. This takes \( \log_2 6429 \) or 13 bits. Once \( k \) is known to both sender and receiver it can be used in the next stage of coding. For example, suppose we view a string \( x^n = (x_1, \ldots, x_n) \in \{0, 1\}^n \) as 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 \( k \) is known, giving us a total code length of

\[
\log_2 n + \left[ -k \log_2 (k/n) - (n - k) \log_2 (1 - k/n) \right].
\] (5)

Under this scheme, we pay 6441 (> 6429) bits to encode the up’s and down’s of the return series, but only 5789 (< 6429) bits for the volatility series. Therefore, relative to sending this information directly, we incur an extra cost of 0.2% on the return string, but save 10% on the volatility string.

From a modeling point of view, we could say that an iid Bernoulli model is postulated for compression or description of a given string and that the Bernoulli probability \( p \) is estimated by \( k/n \). The first term in (5) is the code length for sending \( k \) or the estimated \( p \), while 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 (5), we expect some improvement provided \( k/n \) is not too close to 1/2, and this saving should increase with \( n \). When \( k = n/2 \), however,

\[-k \log_2 (k/n) - (n - k) \log_2 (1 - k/n) = n,
\]

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 0.31. For the return string, on the other hand, the proportion of up’s is close to 1/2, so that the second term in (5) is 6428, just one bit shy of \( n = 6429 \). After adding the additional 13 bit cost to transmit \( k \), 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 string consists of iid observations, then by independence we obtain a joint distribution on \( x^n \) which 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.” Therefore, the code length of this (uniform) mixture code is

\[-\log_2 m(x^n) = -\log_2 k!(n-k)! + \log_2 (n+1)!.
\]

In terms of our original series, by using this mixture code we incur a cost of 6434 bits to transmit the return string and 5782 bits for the volatility string. While consistent with our results for two-stage coding, we have saved 7 bits on both sequences.

Although many mixture codes can be created by making different choices for the prior density assigned to \( p \), the distribution \( m(\cdot) \) is only guaranteed to have a closed form expression 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 they have decided on a Bernoulli encoder with \( p = \frac{1}{2} \). 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 6429 days, this would total 6429 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 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, \( k_{t-1} \) up’s 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 \( p_0 = \frac{1}{2} \). If on day \( t \) we see an increase in the return of the Dow-Jones Industrial Average, then the Bernoulli coder with \( p = \hat{p}_{t-1} \) is used at a cost of \( L_t(1) = -\log_2 \hat{p}_{t-1} \). Otherwise, we transmit a 0, requiring \( L_t(0) = -\log_2 (1 - \hat{p}_{t-1}) \) bits. For a string \( x^n = (x_1, \ldots, x_n) \) with \( k \) 1’s and \( (n-k) \) 0’s, the total code length over 6429 days is

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

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

\[q(x^n) = \prod_{t=1}^{n} \hat{p}_{t-1}^{x_t} (1 - \hat{p}_{t-1})^{1-x_t},\]
where
\[ -\log_2 q(x^n) = \sum_{t=1}^{n} L_r(x_t). \]

Rewriting (7), we find
\[
-\log_2 q(x^n) = -\sum_{t=1}^{n} [x_t \log_2 \hat{p}_{t-1} + (1 - x_t) \log_2 (1 - \hat{p}_{t-1})] \\
= -\sum_{t; x_t = 1} \log_2 \hat{p}_{t-1} - \sum_{t; x_t = 0} \log_2 (1 - \hat{p}_{t-1}) \\
= -\sum_{t; x_t = 1} \log_2 (k_{t-1} + 1) - \sum_{t; x_t = 0} \log_2 (t - k_{t-1}) + \sum_{t=1}^{n} \log_2 (t + 1) \\
= -\log_2 k! - \log_2 (n - k)! + \log_2 (n + 1)!
\]

which is exactly the same expression as (6), the description length derived for the uniform mixture code (an unexpected equivalence that we will return to shortly). Therefore, from the previous example, the predictive code lengths are 6434 bits and 5782 bits for the return and volatility strings, respectively. 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. Clearly, predictive coding requires an ordering of the data which 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. The somewhat surprising equivalence between predictive and mixture code lengths is proved 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 non-stationarity in the data source, a tremendous advantage over the other schemes discussed so far. For example, suppose that we only use the number of up’s encountered in the last 1000 days to estimate \( p \) in a Bernoulli model for the next day’s indicator. When applied to the volatility series, we save only 3 bits over the 5782 needed for the simple predictive coder, implying that this string is fairly stationary. To explore the possible dependence structure in the volatility string, we postulated a first-order Markov model, estimating the transition probabilities from the indicators for the last 1000 days. Under this scheme, we incur a cost of 5774 bits. 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. (In fact, the correlation between \( x_t \) and \( x_{t+1} \) is \(-0.02\), practically non-existent.)

The string derived from the return series, however, is a different story. As with the volatility string, estimating \( p \) based on the previous 1000 days’ data does not result in a smaller code length, suggesting little non-stationarity. However, unlike the volatility string, there is considerably more dependence in this series. The correlation between \( x_t \) and \( x_{t+1} \) for the return string is \(-0.24,\)
indicating that our Markov model might be more effective here. In fact, by postulating a first-order Markov model (estimating transition probabilities at time \( t \) from all the previous data), we reduce the code length to 6181, a 4\% or 253 bit saving over the 6434 bits required for the simple predictive coder. By instead estimating the transition probabilities from the last 1000 days of data, we can produce a further decrease of only 10 bits, confirming our belief that the return string is fairly stationary.

In general, predictive coding can save in terms of code length even when we are considering an iid model. When dependence or non-stationarity are present, we can experience even greater gains by directly modeling such effects, say through a Markov model. Of course, the two-stage and mixture coding schemes can also incorporate these features, and will result in similar code length reductions when the data support the added structure.

### 2.3 The MDL 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. While our focus was on compression, motivation for the MDL principle can be found throughout Example 3: Probability models or descriptions 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. Because the details of specific coding schemes are somewhat unimportant from a methodological standpoint, we will use the terms code length and description length interchangeably when referring to the complexity of a model.

In parametric statistics, we want to estimate the parameter \( \theta \) of a given model (class)

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

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) 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 Shannon’s Source Coding Theorem we recall that the best description length of a data string (in an average sense) given a density \( f_\theta \) in the parametric family \( M \) is simply \(- \log f_\theta(x^n)\) because on average this code achieves the entropy lower bound (3). Therefore, the MDL principle seeks a model that minimizes this description length among all the densities in the family. Obviously minimizing \(- \log_2 f_\theta(x^n)\) is the same as maximizing \( f_\theta(x^n) \), and hence MDL coincides with ML in parametric estimation problems. Therefore, in this setting MDL enjoys all of the desirable properties of ML mentioned above.\(^3\)

It is well known, however, that maximum likelihood breaks down when we are forced to choose among nested classes of parametric models. This occurs most noticeably in variable selection for linear regression.

\(^3\)The quantity \(- \log_2 f_\theta(x^n)\) based on the maximum likelihood estimator \( \hat{\theta}_n \) is only a code length up to an additive constant. As we will see, this constant in the parametric setting depends only on the dimension \( k \) of \( \theta \) so it can be ignored.
The simplest and most illustrative selection problem of this type can be cast as an exercise in hypothesis testing:

**Example 4** Assume \( x^n = (x_1, \ldots, x_n) \) are \( n \) iid observations \( N(\theta, 1) \) for some \( \theta \in \mathbb{R}^1 \), and we want to test the hypothesis \( H_0 : \theta = 0 \) versus \( H_1 : \theta \neq 0 \). Equivalently, we want to choose between the models

\[
M_0 = \{ N(0, 1) \} \quad \text{and} \quad M_1 = \{ N(\theta, 1) : \theta \neq 0 \}
\]

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 \( M_1 \) is always chosen unless \( \bar{x}_n = 0 \), an event with probability 0 even when \( M_0 \) is true.

Notice that ML has no problem with the estimation of \( \theta \) if we merge the two model classes \( M_0 \) and \( M_1 \). It is clear that 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 to each model class, the Bayesian appeals to the posterior probabilities of these classes to select a model. However, the choice of the prior masses is a subjective matter, which in recent years has been made increasingly on the basis of computational efficiency. Another popular Bayesian solution involves the use of BIC, an approximation to the posterior distribution on model classes derived by Schwarz (1978) that removes the prior distribution entirely.

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 \( L_M \) denote the description length of a data string \( x^n \) based on a single family or model (class) \( M \). Because MDL coincides with ML when choosing among members of \( M \), we can think of \( 2^{-L_M} \) 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.

As we have seen in Example 3, however, there are several different coding schemes that can be used to define the description length \( L_M \) of a given model class \( M \). To apply the MDL principle to model selection, we must first decide which prefix codes provide us with “valid” description lengths based on \( M \). 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. In the search for the best description length, Rissanen’s (1986a) lower bound on the redundancy for parametric families is a landmark. Roughly, the expected redundancy of a code corresponds to the price one must pay for not knowing which member of the model class generated the data \( x^n \). Rissanen (1986a) demonstrates that for a regular parametric family of dimension \( k \), this amounts to at least \( \frac{k}{2} \log n \) extra bits. Any code length that achieves this lower bound qualifies (to first order in the parametric case) as a description length of the model class given a data string \( x^n \). An alternative solution to defining the best description length comes from a minimax lower bound on redundancy (Clarke and Barron, 1990). Both of these bounds not only necessitate the use of MDL in statistical model selection problems, but also extend Shannon’s Source Coding Theorem to 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 bounds that $-\log f_{\hat{\theta}}(x^n) + \frac{1}{2}\log n$ is a valid code length for our parametric family introduced at the beginning of this section. Obviously, this takes the same form as BIC. As will be seen throughout the following sections, there are many more connections between forms of MDL and Bayesian schemes. However, there is a fundamental difference between the underlying philosophies motivating each. The description motivation in MDL is firmly grounded in the framework of coding theory, and the universal coding theorems are called upon for its justification. Moreover, MDL allows the use of a description or code length which does not necessarily correspond to a proper distribution as required in the Bayesian approach. This enables MDL to be used in situations where it is difficult to think of a prior even for the practical purpose of regularization, but a sensible code is easy to employ.

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. The description lengths discussed here will be used in our implementation of MDL for the model selection problems in Sections 4 and 5. Three of these schemes were introduced in Example 3 for compression purposes. In that case, probability models helped us build codes that could be employed 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.

As with any new intellectual enterprise, MDL has close ties with existing ideas. As we have seen in Example 3, the different forms of MDL formally share many aspects of both frequentist and Bayesian approaches to model selection. However, the coding framework underlying MDL allows for the development of new forms of description which in turn yield new model selection criteria. For example, at the end of this section we discuss an emerging form of description termed Normalized Maximized Likelihood (NML), which finds motivation in coding theory rather than in problems of statistical inference (Rissanen, 1996, and Barron, Rissanen and Yu, 1998). There is no doubt that other forms of MDL will appear as it is applied to more and more challenging model selection problems.

Throughout this section, we will focus mainly on a simple parametric model class $\mathcal{M}$ consisting of 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 will 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, we would also have to encode information about $\mathcal{M}$, adding to our description length. Finally, we will calculate code lengths using the natural logarithm, rather than $\log_2$ as we did in the previous section. The unit of length is now referred to as a nat.
In the next few pages, we revisit the three coding schemes introduced briefly in Example 3. 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 as well as the minimax lower bound to be covered in Section 5 (Clarke and Barron, 1990). This implies that these schemes produce valid description lengths, each yielding a useable 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 $M$ 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 $\theta$, and then transmitting $\hat{\theta}_n$ with a uniform encoder. Rissanen (1983, 1989) shows 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 will pay a total of $-k \log \frac{1}{\sqrt{n}} = \frac{k}{2} \log n$ nats to communicate an estimated parameter $\hat{\theta}_n$ 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 with the same precision $\delta_m = 1/\sqrt{n}$:

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

where $[\hat{\theta}_n]_{\delta_m}$ is $\hat{\theta}_n$ truncated to precision $\delta_m$. In the MDL paradigm, the distribution $w$ is introduced as an ingredient in the coding scheme and not as a Bayesian prior. However, if we have reason to believe that a particular prior $w$ reflects the likely distribution of the parameter values, choosing $w$ for description purposes is certainly consistent with Shannon’s Source Coding Theorem. It is clear that 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 \( \delta_d \) (\( d \) for the data). In this stage, we can take \( \delta_d \) to be machine precision. The description length for coding \( x^n \) is then
\[
-\log f(x_1, \ldots, x_n|\hat{\theta}_n) - n \log \delta_d.
\]

When the likelihood surface is smooth as in regular parametric families, the difference
\[
\log f(x_1, \ldots, x_n|\hat{\theta}_n) - \log f(x_1, \ldots, x_n|\hat{\theta}_n)
\]
is of a smaller order of magnitude than the model description length \( \frac{k}{2} \log n \). 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 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
\[
-\log f(x_1, \ldots, x_n|\hat{\theta}_n) + \frac{k}{2} \log n. \tag{8}
\]
Again, the first term represents the number of nats needed to encode the date sequence \( x^n \) given an estimate \( \hat{\theta}_n \), while the second term represents the number of nats required to encode the \( k \) components of \( \hat{\theta}_n \) to precision \( 1/\sqrt{n} \). The simplified two-stage description length is valid even if one starts with a \( 1/\sqrt{n} \)-consistent estimator other than the MLE. 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
\[
L_0(x^n) = \frac{1}{2} \sum_{i=1}^{n} x_i^2 + \frac{n}{2} \log(2\pi).
\]
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 \( \theta \) by the MLE \( \hat{\theta}_n = \bar{x}_n \), the two-stage description length (8) takes the form
\[
L_1(x^n) = \frac{1}{2} \sum_{i=1}^{n} (x_i - \bar{x}_n)^2 + \frac{n}{2} \log(2\pi) + \frac{1}{2} \log n. \tag{9}
\]
Therefore, 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 zero as \( n \) tends to infinity. \( \square \)
3.2 Mixture MDL 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). Since 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 above. The proof of this fact (Clarke and Barron, 1990) mimics the original derivation of BIC as an approximate Bayesian model selection criterion (Schwarz, 1978, and Kass and Raftery, 1995). An alternative approximation yields yet another form of description length known as Stochastic Information Complexity (SIC). As we will see, 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 will see 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 that is obtained by taking a mixture of the members in the family with respect to a distribution \( w \) on the parameters:

\[
m(x^n) = \int f_\theta(x^n)w(\theta)d\theta. \tag{10}
\]

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 \( M \). Given a precision \( \delta_d \), 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

\[
- \log \int f(x_1, \ldots, x_n|\theta)w(\theta)d\theta.
\]

This integral has a closed form expression when \( f(\cdot|\theta) \) is an exponential family and \( w \) is a conjugate prior, as was the case in Example 3. 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 the use of Markov chain Monte Carlo (Kass and Raftery, 1995), which can therefore 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 \( \theta \) in \( M_1 \), we find

\[
- \log m(x^n) = \frac{n}{2}\log(2\pi) + \frac{1}{2}\log \det(I_n + \tau J_n) + \frac{1}{2}x_n'(I_n + \tau J_n)^{-1}x_n \tag{11}
\]

where \( I_n \) is the \( n \times n \) identity matrix, and \( J_n \) is the \( n \times n \) matrix of 1’s. Simplifying the above expression, we arrive at

\[
\frac{1}{2} \sum_i x_i^2 - \frac{1}{2} \times \frac{n}{1 + 1/(n\tau)} \bar{x}_n^2 + \frac{n}{2} \log(2\pi) + \frac{1}{2} \left(1 + \frac{1}{n\tau}\right) \log(1 + n\tau) \tag{12}
\]
Figure 3: Comparing the penalties imposed by BIC and the mixture form of MDL for $\tau = 0.5$ and $\tau = 2$. The sample size $n$ ranges from 1 to 50.

Comparing this to the description length for the two-stage encoder (9), we find a difference in the penalty

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

which (to first order) is asymptotically the same as that associated with BIC, $\frac{1}{2} \log n$. Depending on the value of the prior variance $\tau$, the quantity (13) represents either a heavier ($\tau > 0$) or a lighter ($\tau < 1$) penalty. In Figure 3 we present a graphical comparison for two values of $\tau$.

An analytical approximation to the mixture $m(\cdot)$ in (10) 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 the Stochastic Information Complexity:

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

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 $n \to \infty$,

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

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

$$\frac{1}{2} \log \det(nI(\hat{\theta})).$$
can be interpreted as the number of nats needed to encode the \( k \) estimated parameter values if we discretize the \( j \)th parameter component with a precision \( \text{SE}(\hat{\theta}_j) = 1/\sqrt{n\lambda_j(0)} \) (provided the estimated parameters are either independent or the discretization is done after transforming the parameter space so that the information matrix under the new parametrization is diagonal). It is obviously sensible to take into account the full estimation error when discretizing, and not just the rate. The final equality in (15) tells us that in the limit, \( \text{SIC} \) is approximately \( \text{BIC} \) or two-stage MDL. For finite sample sizes, however, \( \text{SIC} \)'s penalty term is usually not as severe as \( \text{BIC} \)'s, and hence in some situations, \( \text{SIC} \) outperforms \( \text{BIC} \). Rissanen (1989, pp. 151, Table 6) illustrates this difference by demonstrating that \( \text{SIC} \) outperforms two-stage MDL when selecting the order in an AR model with \( n = 50 \). In Section 4, we will 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, each of the conditionals \( f_\theta(x_j|x_1, \ldots, x_{j-1}) \) share the same parameter \( \theta \).\footnote{Typically, \( f(x_1) = f_0(x_1) \) will not depend on \( \theta \), 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

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

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 (17) is

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

The MDL model selection criterion based on this form of description is called \text{PMDL} for its use of the predictive distribution (17).

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 pre-determined encoder \( f_0 \) to transmit the first data point \( x_1 \). This accounts for the leading term in the summation (18). 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 \( t \)th term in the summation (18). 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

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

(with $\bar{x}_0 = 0$) based on $M_1$, then

$$-\log q(x^n) = -\sum_{t=1}^{n} \log f_{\hat{\theta}_{t-1}}(x_t|x^{t-1})$$

$$= \frac{n}{2} \log(2\pi) + \frac{1}{2} \sum_{t=1}^{n} (x_t - \bar{x}_{t-1})^2. \quad (19)$$

The reasoning we followed in deriving PMDL is identical to the prequential approach to statistics advocated by Dawid (1984, 1991). The form (18) 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 Rissanen (1986b), Hemery and Davis (1989), Hannan and Kavalieris (1984), Hannan, McDougall and Poskitt (1989), Hannan and Rissanen (1982), Gerencsér (1994), Wei (1992), and Speed and Yu (1994). The recent results of Qian, Gabor and Gupta (1996) extend the horizon of this form of MDL to generalized linear models.

In Section 4, we will illustrate the application of PMDL to the (differenced) daily return series studied in Example 3. In this case we will 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) suggests repeatedly permuting the data before applying PMDL, and then averaging the predictive code lengths. In Section 4, we avoid these complications and only discuss PMDL in the context of time series data.

### 3.4 Other Forms of Description Length

The MDL principle offers one the opportunity to develop many other forms of description length, in addition to the three discussed above. 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 all 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 coding scheme of Shtarkov (1987) (cf. Barron et al., 1998). For a flavor of how it was derived, we apply NML (for normalized maximized likelihood) to the binary, Dow-Jones up-and-down indicators introduced in Section 2.
Example 3 (continued) Normalized Maximized Likelihood Coding. As was done in the two-stage scheme, we first transmit $k$. Then, both sender and receiver know that the indicator sequence is in type class $T(n,k)$, the collection of strings of size $n$ with exactly $k$ 1’s. Under the iid Bernoulli model, each string in the class is equally likely, we can employ a uniform code on $T(n,k)$ for communicating its elements. When applied to the return string, the $NML$ code requires $\log_2 \frac{n!}{k!(n-k)!}$ or 6421 bits, giving us a total code length of 6434 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, where $x^n$ 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 example above, this meant coding the return string as an element of $T(n,k)$ rather than $\{0,1\}^n$. Rissanen (1996) formally introduces this scheme for MDL model selection, and discusses its connection with minimax theory. We will see another application of this code when we take up ordinary linear regression in the next section.

4. Applications of MDL in Model Selection

4.1 Linear Regression Models

We begin this section with the simplest class of statistical models, normal linear regression. Our goal is to understand 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, $\gamma_m$, and consider the model

$$y = \sum_{m=1}^{M} \beta_m x_m + \varepsilon,$$

(20)

where $\varepsilon$ has a Gaussian distribution with mean zero and unknown variance $\sigma^2$. The vector $\gamma = (\gamma_1, \ldots, \gamma_M) \in \{0,1\}^M$ will be used as a simple index for the $2^M$ possible models given by (20). In this section, we apply MDL to the problem of model selection, or equivalently, the problem of identifying a vector $\gamma$ that yields the “best” prediction of $y$ in equation (20). When $E(y|x_1, \ldots, x_M)$ is contained in this class of models, we know from Speed and Yu (1993) and Barron et al. (1998) that each of these MDL criteria is consistent, selecting the true value of $\gamma$ with probability approaching one as the number of samples tends to infinity.

Because variable selection in linear regression is an extremely common problem that arises in many applications, we will discuss several MDL criteria in detail. In each criterion, we omit the code length $L(\gamma)$ for encoding the model indexed by $\gamma$. This is justified if we use the same code length $\log_2 2^M = M$ for each model (corresponding to the uniform prior on all models considered). As we have seen previously, two-stage MDL is equivalent to using $BIC$ for model selection. Because many readers will be familiar with this procedure, it (along with $AIC$) will serve as a benchmark throughout our discussion. We will also consider two versions of the mixture form of MDL. The first is $gMDL$, so named because we assign Zellner’s $g$-prior (Zellner, 1986) to the regression coefficients. By choosing an identity prior instead, we obtain a second mixture form, $iMDL$. The use of Zellner’s $g$-prior in this context is new, while Rissanen (1989) discusses $iMDL$. Next, by applying the normalized maximum likelihood code introduced in Section 3.4 to linear
regression we derive $nMDL$. The last three model selection criteria, $gMDL$, $iMDL$, and $nMDL$, each has one or more hyperparameters that we choose by minimizing the resulting code lengths, so that the general consistency results for MDL do not apply. By examining the forms of these criteria, however, consistency can be established directly. Finally, we consider variable selection with an approximate MDL form $SIC$, the stochastic information complexity, introduced at the end of Section 3.2.

Through a small simulation study, we explore the finite sample behaviors of these MDL model selection procedures. We also illustrate their practical application on two data sets. The first comes from an experiment designed to identify genetic markers that influence the number of bristles on the abdomen of a fruit fly (Broman, 1997). The second consists of spectral reflectance measurements obtained from samples of common fruits and vegetables. In the latter application, MDL is used to select knot locations in a polynomial spline model for the reflectance curves as a function of wavelength. The coefficients from this model will be used in Section 4.2 as inputs to a clustering algorithm also based on the MDL principle.

The material in this section draws from a number of sources on MDL (see Rissanen, 1989, Speed and Yu, 1993, and Barron et al, 1998) as well as the current literature on Bayesian variable selection (see George and Foster 1998, Smith and Kohn 1996, and Wong, Hansen, Kohn and Smith 1997). By considering normal linear regression, we hope to highlight both the derivation and application of the MDL principle to model selection.

### 4.1.1 Several Forms of MDL for Model Selection

Most of our readers will be familiar with two-stage MDL or the use of $BIC$ for model selection in the context of simple linear regression. When $\sigma^2$ is unknown, recall that $BIC$ selects the model vector $\gamma$ minimizing

$$BIC(\gamma) = n \frac{1}{2} \log \text{RSS}(\gamma) + \frac{k_\gamma}{2} \log n,$$

where $\text{RSS}(\gamma)$ is the residual sum of squares and $k_\gamma$ is the number of variables in the model corresponding to a particular index $\gamma$. Here, the penalty applied to the dimension $k_\gamma$ depends on the sample size $n$. Related criteria like $AIC$ differ in the form of this penalty:

$$AIC(\gamma) = n \frac{1}{2} \log \text{RSS}(\gamma) + k_\gamma.$$  

Because the space of possible models is enormous for even moderate values of $M$, traditional methods for approximately optimizing (21) or (22) employ either stepwise procedures or some variant of “leaps and bounds” (Furnival and Wilson, 1974). An and Gu (1985) find that the minimum $BIC$ model selected by greedy, stepwise deletion of variables yields a consistent estimate of $\gamma$. Recent advances in Bayesian computation have led to various Markov chain Monte Carlo (MCMC) sampling techniques for identifying promising values of $\gamma$. Samplers specifically aimed at minimizing $BIC$ have been discussed by George and McCulloch (1997).

To implement the mixture form of MDL, we must first choose a distribution on the parameters $\sigma^2$ and $\beta$. Recall that this form is structurally similar to Bayesian analysis, and hence our discussion will borrow from classical Bayesian approaches to linear regression models. To simplify notation, pick a model vector $\gamma$ and let $\beta = \beta_\gamma$ correspond to the coefficients in (20) for which $\gamma_m = 1$. For this fixed choice of $\gamma$, we assign an
inverted gamma distribution to the parameter \( \tau = \sigma^2 \),

\[
p(\tau) = \sqrt{\frac{a}{2\pi}} \tau^{-3/2} \exp\left(\frac{-a}{2\tau}\right),
\]

and a multivariate Gaussian distribution to \( \beta \) with mean 0 and variance-covariance matrix \( c\tau \Sigma \). Clearly, \( \Sigma = \Sigma_\gamma \) depends on the model \( \gamma \), and we will also let the scaling factors \( a = a_\tau \) and \( c = c_\gamma \) vary from model to model. Let \( k = k_\gamma \) denote the number of variables in the model (20), and set

\[
S_c = y' (I_{n \times n} + cX\Sigma X')^{-1} y,
\]

where \( X \) is the \( n \times k \) design matrix corresponding to model \( \gamma \). By integrating with respect to \( \beta \) and \( \tau \), we find that the negative, log-marginal density of \( y \) is given by

\[
-\log m(y) = \frac{1}{2} \log a + \frac{n+1}{2} \log(a + S_c) + \frac{1}{2} \log |I_{n \times n} + cX\Sigma X'|,
\]

up to a universal additive constant depending only on \( n \). For each model, we minimize (23) with respect to \( a \), setting \( \hat{a} = S_c/n > 0 \), to produce the following expression

\[
-\log m(y) = \frac{n}{2} \log S_c + \frac{1}{2} \log |I_{n \times n} + cX\Sigma X'|.
\]

(24)

By choosing the hyper-parameter \( a \) in this way, the MDL approach distinguishes itself from a strict Bayesian analysis. From a coding perspective, however, this technique is natural; we want to construct the shortest possible code for each model. Note that the optimal \( a \) should be transmitted with a fixed high precision, but that this adds an amount to the code length which is independent of the chosen model. Hence, we will omit its contribution in the following expressions derived from (24).

Depending on the problem at hand, there are a variety of choices for \( \Sigma \). As we will see, the most mathematically tractable was proposed by Zellner (1986) who christened it the \( g \)-prior. Specifically, we let \( \Sigma = (X'X)^{-1} \), so that

\[
S_c = \frac{y'y + cRSS}{1 + c},
\]

and

\[
|I_{n \times n} + cX\Sigma X'| = (1 + c)^k.
\]

Hence, the mixture form of MDL with Zellner’s \( g \)-prior for \( \beta \) is given by

\[
-\frac{n-k}{2} \log(1 + c) + \frac{n}{2} \log(y'y + cRSS).
\]

(25)

Rather than optimizing for \( a \) as we have done, Smith and Kohn (1996) began with an improper prior for \( \tau \) and obtain (25) as the (negative) log-posterior of a model index \( \gamma \) in the context of Bayesian variable selection for the normal linear model (see also George and Foster, 1998). While we arrive at the same expression, improper priors do not make sense from a coding perspective.
By taking $\Sigma = (X'X)^{-1}$ we can derive in closed-form the value of $\hat{c} = \hat{c}_g$ that minimizes (25), namely

$$
\hat{c} = \max(F - 1, 0)
$$

(26)

where $F$ is the usual $F$-ratio for testing the hypothesis that each element of $\beta$ is zero,

$$
F = \frac{(y'y - \text{RSS})}{kS},
$$

and $S = \text{RSS}/(n - k)$. Rewriting expression (26), we find that $\hat{c}$ is zero unless $R^2 > k/n$, where $R^2$ is the usual squared multiple correlation coefficient. Substituting the optimal value of $\hat{c}$ into (25), we arrive at a mixture form of MDL which we will refer to as $gMDL$ for its use of the $g$-prior:

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

(27)

For any model satisfying $R^2 \leq k/n$, the optimal value of $c$ is zero and the prior on $\beta$ becomes a point mass at zero. The $gMDL$ criterion equates all such degenerate $\gamma$ with the null model. Since $n/n = 1 \geq R^2$, this version of mixture MDL with Zellner’s $g-$prior for $\beta$ can never choose a model with dimension larger than the number of observations. When implementing this criterion, it is preferable to discretize $\hat{c}$ so that exact zeros are not possible. This kind of adjustment is consistent with the coding framework under which $\hat{c}$ has finite precision. George and McCulloch (1997) discuss similar restrictions in the context of an empirical Bayes procedure based on the practical significance of the coefficients.

It can be shown that for a fixed dimension $k$, $gMDL$ will select the lowest RSS model, a property shared by $BIC$ and $AIC$ as well as other selection criteria like Mallows’ $C_p$ (Mallows, 1973). Among models with the same RSS, the criterion favors the model with the smallest dimension $k$. In general, because the value of $\hat{c}$ increases in $R^2$ and decreases in $k$, small models that fit the data well are assigned large values of $\hat{c}$. Big values of $\hat{c}$ correspond to a diffuse prior for $\beta$. On the other hand, large poor-fitting models receive a small value of $\hat{c}$, which corresponds to a prior for $\beta$ concentrated at zero.

The use of Zellner’s $g$-prior is convenient precisely because there exists a closed-form expression for the mixture $gMDL$ criterion (27) after optimizing over $c$. Another popular choice for the prior variance-covariance matrix $\Sigma$ is the identity $I_{k \times k}$. Rissanen (1989) discusses this option and derives the resulting mixture MDL criterion. Replacing $\Sigma$ by $I_{k \times k}$ in (24) yields

$$
- \log m(y) = \frac{n}{2} \log S_c + \frac{1}{2} \log \left[ \text{RSS} + cXX' \right],
$$

with $S_c = y'(I + cXX')^{-1}y$. To identify a value of $c$ that minimizes the above expression, we introduce a new parameter $t = 1/c$, and let $\text{RSS}_t$ denote the residual sum of squares corresponding to the ridge regression problem

$$
\text{RSS}_t = \min_{\beta} \left( \text{RSS}(\beta) + t\beta'\beta \right),
$$

(28)
where \( RSS(\beta) = \| y - X\beta \|^2 \). Then, it is straightforward to verify that the optimal value of \( t \) satisfies

\[
 t = \frac{kRSS_y}{\text{RSS} + \text{trace}\left[(nI_{k \times k} + X'X)^{-1}\right] + ny'X(nI_{k \times k} + X'X)^{-1}X'y}.
\]  

(29)

This expression can be be applied iteratively, with convergence typically requiring fewer than twenty steps, depending on the starting values. Rissanen (1989, pp. 129) presents a slightly different relationship, presumably the result of transcription errors. It is conceivable that iterations of (29) can lead to an optimizing value of \( \hat{t} = \infty \) or \( \hat{c} = 0 \) for certain \( \gamma \) as we observed in the derivation of gMDL. However, it is difficult to specify conditions under which this should occur and we have not encountered the problem in practice.

Substituting the minimizing value of \( \hat{c} = 1/\hat{t} \) obtained from successive iterations of (29), we arrive at the mixture form of MDL which we will refer to as \( iMDL \) for its use of the identity or independence prior:

\[
iMDL = \frac{n}{2} \log S_\xi + \frac{1}{2} \log |I_{n \times n}| + \hat{c}XX'.
\]  

(30)

In Section 3.2, we applied a simple approximation to the mixture form of MDL to derive the so-called Stochastic Information Complexity (14). For a model index \( \gamma \), the Hessian matrix of the mixture \( m(\cdot) \) in (10) based on the \( k + 1 \) parameters \( \beta \) and \( \tau = \sigma^2 \) is given by

\[
\begin{pmatrix}
\frac{k}{2}X'X & 0 \\
0 & \frac{n}{2\tau}
\end{pmatrix}.
\]

Therefore, a little algebra reveals the SIC criterion

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

(31)

where we have omitted an additive constant that is independent of \( \gamma \).

As mentioned in Section 3.4, the normalized maximum likelihood form of MDL (cf. Rissanen, 1996, and Barron et al., 1998) is very recent and only some of its theoretical properties are known. It is motivated by the maximum likelihood code introduced by Shtarkov (1987). Recall that based on a model with index \( \gamma \), the maximum likelihood estimates of \( \hat{\beta} \) and \( \hat{\tau} = \sigma^2 \) are

\[
\hat{\beta}(y) = (X'X)^{-1}X'y \quad \text{and} \quad \hat{\tau}(y) = ||y - X\hat{\beta}||^2 / n.
\]

Let \( f(y|X, \beta, \tau) \) be the joint Gaussian density of \( y \) corresponding to the model \( \gamma \), so that the normalized maximum likelihood function is

\[
\hat{f}(y) = \frac{f(y|X, \hat{\beta}(y), \hat{\tau}(y))}{\int_{\mathcal{Y}(\tau, \tau_0)} f(z|X, \hat{\beta}(z), \hat{\tau}(z))dz},
\]

where \( \mathcal{Y}(\tau, \tau_0) = \{ z|\hat{\beta}'(z)X'X\hat{\beta}(z)/n \leq \tau, \hat{\tau}(z) \geq \tau_0 \} \). Using the sufficiency and independence of \( \hat{\beta}(y) \) and \( \hat{\tau}(y) \), one obtains

\[
-\log \hat{f}(y) = \frac{n}{2} \log RSS - \log \Gamma\left(\frac{n-k}{2}\right) - \log \Gamma\left(\frac{k}{2}\right) + \frac{k}{2} \log \frac{r}{\tau_0} - 2\log(2k).
\]  

(32)

23
To eliminate the hyper-parameters $r$ and $\tau_0$, we again minimize the above code length for each model $\gamma$ by setting

$$r = \frac{\hat{\beta}(y)X'X\hat{\beta}(y)}{n} = \frac{y'y - RSS}{n} \quad \text{and} \quad \tau_0 = \hat{\tau}(y) = \frac{RSS}{n}.$$ 

By substituting these values into (32), we obtain the selection criteria $nMDL$ ($n$ for “normalized maximum likelihood”),

$$nMDL = \frac{n}{2} \log RSS - \log \Gamma \left( \frac{n-k}{2} \right) - \log \Gamma \left( \frac{k}{2} \right) + \frac{k}{2} \log \frac{y'y - RSS}{RSS} - 2\log(2k). \quad (33)$$

Rewriting this expression, we find that

$$nMDL = \frac{n}{2} \log S + \frac{k}{2} \log F$$

$$+ \frac{n-k}{2} \log(n-k) - \log \Gamma \left( \frac{n-k}{2} \right) + \frac{k}{2} \log(k) - \log \Gamma \left( \frac{k}{2} \right) - 2\log k,$$

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

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

We recognize the leading two terms in this expression as the value of $gMDL$ (27) when $R^2 > k/n$. This structural similarity is interesting given that these two MDL forms were derived from very different codes.

### 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 residual sum of squares, $RSS$. Therefore, to implement these criteria, it is sufficient to consider only the lowest $RSS$ models for dimensions $1, 2, \ldots, M$. Because the number of predictors in our simulations is relatively small, it is not unreasonable to perform an exhaustive search for these models by a routine branch-and-bound algorithm. 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) discusses 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 (10) that we can write as $m(y) = p(y|\gamma)$ for model indices $\gamma$. 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. In what follows, we will estimate the posterior mode of $\gamma$ (assuming the stochastic model implied by either $SIC$ or $iMDL$) using a variant of the Gibbs
Table 1: Simulation results for \( n = 20 \) observations from model (34). In each case, \( \rho = 0.5 \) and \( \sigma = 4.0 \).

<table>
<thead>
<tr>
<th>( \beta = (5,0,0,0,0,0,0,0) )</th>
<th>( \text{OLS} )</th>
<th>Median model error</th>
<th>Average model size</th>
<th>Proportion correct</th>
<th>Equivalent penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>(snr ( \approx 3.2 ))</td>
<td>10.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>gMDL</td>
<td>1.6</td>
<td>1.6</td>
<td>0.6</td>
<td>4.0</td>
<td></td>
</tr>
<tr>
<td>nMDL</td>
<td>5.7</td>
<td>2.7</td>
<td>0.2</td>
<td>2.4</td>
<td></td>
</tr>
<tr>
<td>iMDL</td>
<td>2.1</td>
<td>1.7</td>
<td>0.6</td>
<td>3.7</td>
<td></td>
</tr>
<tr>
<td>BIC</td>
<td>4.4</td>
<td>2.3</td>
<td>0.4</td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>AIC</td>
<td>7.2</td>
<td>3.1</td>
<td>0.2</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td>SIC</td>
<td>8.2</td>
<td>3.8</td>
<td>0.01</td>
<td>1.1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \beta = (3,1.5,0,0,2,0,0,0) )</th>
<th>( \text{OLS} )</th>
<th>Median model error</th>
<th>Average model size</th>
<th>Proportion correct</th>
<th>Equivalent penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>(snr ( \approx 3.2 ))</td>
<td>9.8</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>gMDL</td>
<td>5.9</td>
<td>2.6</td>
<td>0.2</td>
<td>3.6</td>
<td></td>
</tr>
<tr>
<td>nMDL</td>
<td>7.4</td>
<td>3.2</td>
<td>0.2</td>
<td>2.5</td>
<td></td>
</tr>
<tr>
<td>iMDL</td>
<td>6.0</td>
<td>2.8</td>
<td>0.2</td>
<td>2.8</td>
<td></td>
</tr>
<tr>
<td>BIC</td>
<td>7.4</td>
<td>3.0</td>
<td>0.1</td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>AIC</td>
<td>8.3</td>
<td>3.6</td>
<td>0.1</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td>SIC</td>
<td>8.5</td>
<td>5.2</td>
<td>0.04</td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \beta = 0.75 \ast (1,1,1,1,1,1,1,1) )</th>
<th>( \text{OLS} )</th>
<th>Median model error</th>
<th>Average model size</th>
<th>Proportion correct</th>
<th>Equivalent penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>(snr ( \approx 1.4 ))</td>
<td>11.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>gMDL</td>
<td>10.2</td>
<td>3.0</td>
<td>0.0</td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>nMDL</td>
<td>9.6</td>
<td>4.0</td>
<td>0.0</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td>iMDL</td>
<td>9.4</td>
<td>3.5</td>
<td>0.0</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td>BIC</td>
<td>10.4</td>
<td>3.1</td>
<td>0.0</td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>AIC</td>
<td>9.7</td>
<td>3.8</td>
<td>0.0</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td>SIC</td>
<td>10.3</td>
<td>4.9</td>
<td>0.1</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

The so-called focused sampler is relatively easy to implement and has been shown to perform favorably even in high-dimensional variable selection problems with \( k \approx n \). Reasonable alternatives 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).

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

\[
y = x\beta + \sigma \varepsilon,
\]

where \( x \in \mathbb{R}^8 \) has a multivariate normal distribution with mean zero and variance-covariance matrix \( V_{ij} = 2\rho|i-j|, i, j = 1, \ldots, 8; \) and \( \varepsilon \) is an independent standard normal noise term. In Table 1, we compare several MDL selection criteria across 100 data sets simulated according to (34), where \( \rho = 0.5, \sigma = 4 \) 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 to be

\[ E\{x\hat{\beta} - x\beta\}^2 = (\hat{\beta} - \beta)^\intercal V (\hat{\beta} - \beta), \]

with \( \hat{\beta} \) obtained by an ordinary least squares (OLS) fit with the selected variables. In Table 1 we have also included 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 general, except for SIC, the MDL criteria outperformed AIC and BIC. In only one of the simulation suites did gMDL perform poorly, namely the third case with entirely weak effects. When we increase the sample size to 50, but maintain the same signal-to-noise ratio, gMDL recovers and its model error rivals that of iMDL. This breakdown for small sample sizes appears to be a problem with gMDL’s implicit estimation of \( \Sigma \) through the use of Zellner’s prior. Another interesting effect to mention in Table 1 is that in the third case (weak effects), model selection with iMDL out-performs OLS and AIC. In principle, AIC is known to work well in this situation. When we re-ran 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 (28) is responsible for iMDL’s excellent performance here. We hasten to add, however, that once a model is selected, we are simply performing an OLS fit to obtain \( \hat{\beta} \). In both mixture forms of MDL, the shrinkage estimates based on \( \hat{c} \) or \( \hat{i} \) improve on these OLS estimates. The interested reader is referred to Hansen and Yu (1998).

Given the penalties on \( k \) imposed by AIC and BIC, one can expect that AIC will favor larger models while BIC is more conservative. This can be seen in each of our simulation results. The MDL forms, however, 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, consider the plot of say iMDL versus model size (Figure 4). Define iMDL\(^*\) (\( k \)) to be the minimum value of iMDL among all models of size \( k \) and let RSS\(^*\) (\( k \)) be the residual sum of squares for that model. Then, consider the quantity

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

If we replaced iMDL with either AIC or BIC in this definition, then the difference \( \lambda(k + 1) - \lambda(k) \) would be 2 or \( \log n \), respectively.\(^5\) 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 (see Figure 4). 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 simulations. When faced with a single, strong effect, for example, the penalty associated with iMDL is larger than that of BIC, forcing smaller models; while when given a number of small effects, the penalty shrinks below that for AIC allowing iMDL to capture larger models. The criterion SIC tends to impose a penalty that is even weaker than AIC, leading to its discouraging results.

\(^5\)While the expressions for BIC and AIC 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.
Figure 4: 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.

From these simulations, we find that there is a distinct performance advantage in the adaptive forms of MDL, $gMDL$ and $iMDL$, over BIC or AIC in model selection. As $gMDL$ and $iMDL$ are new, their behaviors must be studied more extensively (Hansen and Yu, 1998). Interestingly, both 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, a hand-crafted procedure has been proposed to perform model selection within a restricted class of problems. We find that the adaptivity of MDL produces results that are equivalent to this specialized approach. In the second example, we apply MDL to curve estimation. The output from this procedure will be used later to illustrate a form of MDL for cluster analysis.

4.1.3 Applying MDL 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, Mullaney, Reid, Fry, Langley and Mackay (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 will
attempt to distill the essential features. First, a sample of fruit flies were selectively inbred to produce two family lines differentiated on the basis of their abdominal bristles. Those flies with low bristle counts were separated into one parental line L, while 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 F₁, and then the F₁ flies were again crossed with the low parental line L. Ultimately, sixty-six inbred family lines were obtained in this way so that the individual flies within each group were genetically identical at nineteen 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 the flies bred in the backcross inherited one chromosome from the first filial generation F₁ and one from the low parental line L, so that at each of the genetic markers they have either the LL or HL genotype. The goal of this experiment was to identify whether the genotype at any of the nineteen 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

\[
y_{ij} = \mu + \alpha s_j + \sum_l \beta_l x_{il} + \sum_l \delta_l s_j x_{il} + \epsilon_{ij} \tag{35}
\]

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 \( l \)th marker, \( l = 1, \ldots, 19 \). Therefore, the full model (35) includes main effects for sex and genotype as well as the complete sex × genotype interaction, a total of 39 variables. The error term \( \epsilon_{ij} \) is taken to be Gaussian with mean zero and unknown variance \( \sigma^2 \). In this framework, identifying genetic markers that have an influence on bristle counts becomes a problem of selecting genotype contrasts in the model (35). 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. Therefore, in the notation of Section 4.1 we introduce an index vector \( \gamma \in \{0, 1\}^{39} \) that determines which covariates in (35) 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

\[
BIC_\eta = \frac{n}{2} \log \text{RSS}(\gamma) + \eta \frac{k_\gamma}{2} \log n, \tag{36}
\]

where \( \eta = 2, 2.5, \text{ or } 3 \). 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 (36) by appealing to the framework in Smith (1996) and Smith and Kohn (1996) which led to our expression (25) in Section 4.1.1. If instead of finding optimal values for \( c_\gamma \), we simply fix \( c = c_\gamma = n^\eta \) for all models \( \gamma \), then from (25) we roughly obtain Broman’s criterion (36).⁶ The larger we make \( \eta \), the more diffuse our prior on \( \beta_\gamma \) 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.

⁶This argument is meant as a heuristic; for the precise derivation of (36), the interested reader is referred to Broman (1997).
Comparing several forms of MDL

As the number of markers studied by Long et al. (1997) was relatively small, Broman (1997) was able to employ a branch-and-bound procedure to obtain the optimal model according to each of the criteria (36). By good fortune, these three rules each selected the same 8-term model,

\[ y_{ij} = \mu + \alpha s_j + \beta_2 x_{i2} + \beta_5 x_{i5} + \beta_9 x_{i9} + \beta_{13} x_{i13} + \beta_{17} x_{i17} + \delta_{Sj} x_{i5} + \epsilon_{ij}, \]  

(37)

which includes the main effect for sex, five genotype main effects (occurring at markers 2, 5, 9, 13, and 17), and one sex × 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, and applied the focused sampler of Wong et al. (1997) to locate promising models for iMDL and SIC. In Figure 5 we overlay these criteria, plotting the minimum of each as a function of the model dimension \( k \).

Since the MDL forms were derived up to additive constants, we have mapped each curve to the interval \([0, 1]\). As noted by Broman (1997), BIC and hence AIC chose larger models that were primarily supersets of (37) involving 9 and 13 terms, respectively. Our two forms of mixture MDL, gMDL and iMDL, and
the Normalized Maximized Likelihood criterion, \( nMDL \), were each in agreement with Broman’s \( BIC_\eta \), selecting model (37). Using the device introduced in the previous section (see Figure 4), we find that the equivalent penalty imposed by \( gMDL \) was 7.4, which corresponds to \( \eta = 7.4 / \log n = 7.4 / \log 132 = 1.5 \). For \( nMDL \) the story was about the same with an equivalent penalty of 7.0 (or an \( \eta \) of 1.4). Finally, \( iMDL \) had a penalty of 6.4 for an \( \eta \) 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 non-spurious 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 the \( SIC \) 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 (36). For the most part, our results are consistent with Broman (1997), identifying (37) 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. In Figure 6 we present a number of spectral reflectance curves obtained from samples of common fruits and vegetables. In total, measurements were taken on samples from some 70 varieties of popular produce, our ultimate goal being the creation of a recognition system that could augment supermarket check-out systems. For example, in the upper lefthand panel, 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 6. 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 zero and variance 1.

To make sense of these curves, consider the sample of limes represented in the upper rightmost corner of Figure 6. 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, 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 we observe each curve at \( n \) distinct wavelengths \( x_1, \ldots, x_n \). Then, consider the candidate basis functions of the form

\[
B_i(x) = K(x, x_i) \quad \text{for} \quad i = 1, \ldots, n,
\]
Figure 6: Spectral reflectance curves collected from 6 varieties of supermarket produce. In each panel, we plot 5 representative curves.
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 and Wong et al., 1997). Then, having settled on a basis, we search for an approximation of the form

$$f(x) \approx \alpha_0 + \alpha_1 x + \sum_{i=1}^n \beta_i B_i(x), \quad x \in [400, 800]$$

(38)

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 (38) with $B_i$ defined through smoothing spline kernels is equivalent to choosing knot locations in a natural spline space (Schumaker, 1981). Notice that in this case we always include a constant and linear term in our fits. (Because of our normalization, we do not need the constant term, but we include it in the equation above for completeness). In this context, Luo and Wahba (1997) employ a stepwise greedy algorithm to identify a model $\gamma$, while Wong et al. (1997) make use of the focused sampler after constructing a computationally feasible prior on $\gamma$. Finally, recall that a traditional smoothing spline estimate would fix $\gamma = (1, \ldots, 1)$ and perform a penalized fit (Wahba, 1990). See Hansen and Kooperberg (1998) for a general discussion of knot location strategies.

As mentioned above, the data presented in Figure 6 was collected as part of a larger project to create a classifier for recognizing supermarket produce based solely on its color. While we ultimately applied a variant of penalized discriminant analysis (Hastie, Buja and Tibshirani, 1995), a reasonably accurate scheme involves dimension reduction (38) followed by simple linear discriminant analysis (LDA) on the coefficients $\beta_i$. Therefore, we adapted the MDL criteria introduced previously to handle multiple responses (curves). Our search for promising indices $\gamma$ now represents identifying a single spline space (38) 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 (38), 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 listed in Figure 6 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 (29) required by iMDL are prohibitive given our current implementation of the algorithm. It is of course possible to take short-cuts with greedy, deterministic searches as proposed by Rissanen (1989). However, to simplify our presentation, we restrict our attention to only these two forms. In each case, 10,000 iterations of the sampler were used to identify the best expansion (38). 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. In Figure 7 we highlight 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 (38). 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 shift.

Based on these selected knot locations, we now project each curve into the linear space defined in (38). In the next section, the coefficients from these projections will be applied to an 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 mentioned 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; and
- 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 we 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 will attempt 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 has to do with 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 (38) with the 14 knot locations specified in Figure 7. When combined with the linear term in (38) 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 www.cs.monash.edu.au/~dld. Wallace and Dowe (1994) describe 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 intra-class correlations are zero. Following a suggestion in the documentation, we orthogonalized the entire data set via a principal components decomposition. In Figure 8, have plotted 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 cantelope 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 cantelope 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), while in other cases the color appears in too many different varieties.

### 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 autoregressive-moving average (ARMA) process is a common model selection problem. Throughout this section we will focus on Gaussian ARMA($p, q$) models, specified by the equation

$$x_t = \phi_1 x_{t-1} + \ldots + \phi_p x_{t-p} + Z_t + \theta_1 Z_{t-1} + \ldots + \theta_q Z_{t-q},$$

(39)

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

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

have no roots in $|z| < 1$, so that equation (39) 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 (39) to times $t+1, t+2, \ldots$ conditional on the first $t$ data points. For example,
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. The symbols denote B = Banana, Li = Lime, Le = Lemon, C = Cantelope, O = Orange, and G = Garlic.
following Rockwell and Davis (1991, pp. 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 formulae:

$$
\begin{align*}
\hat{x}_{t+1} &= \sum_{i=1}^{t} \theta_i (x_{t+i} - \hat{x}_{t+i}), \\
\hat{x}_{t+1} &= \phi_1 x_t + \ldots + \phi_p x_{t-p} + \sum_{i=1}^{q} \theta_i (x_{t+i} - \hat{x}_{t+i}),
\end{align*}
$$

1 \leq t < \max(p, q)

$$
\begin{align*}
\hat{x}_{t+1} &= \phi_1 x_t + \ldots + \phi_p x_{t-p} + \sum_{i=1}^{q} \theta_i (x_{t+i} - \hat{x}_{t+i}), \\
& \quad t \geq \max(p, q)
\end{align*}

(40)

The extra parameters $\theta_i$ and $r_t$ can be obtained recursively by applying the so-called innovation algorithm (Rockwell and Davis, Prop. 5.2.2., 1991) 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 will collect the parameters $\phi, \theta$ and $\sigma^2$ into a single vector $\beta$. To emphasize the dependence of $\hat{x}_{t+1}$ and $r_t$ on $\beta$, we write $\hat{x}_{t+1}(\beta)$ and $r_t(\beta)$.

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

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

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

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

(41)

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 AR models and from Gerenceser (1987) for general ARMA processes. As explained earlier, the complexity penalty $\log n/2$ comes from coding the parameter values at the estimation rate $1/\sqrt{n}$. When an AR model is not stable, Huang (1990) shows that 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

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

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

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

although APE was used long before the MDL principle. The computational cost of $PMDL$ can be enormous for general ARMA models since the parameter estimate $\hat{\beta}_n$ in (42) 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 can be found for AR models in Hannan, McDougall, and Poskitt (1988) and Hemerly and Davis (1989a, 1989b), and for general ARMA models in Gerencser (1987).

While deriving a mixture form of MDL appears possible by appealing to the state-space approach to ARMA processes (cf. 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 the lefthand panel of Figure 2, we presented first differences of the daily return series. While our interest at that point was on compressing the string of up’s and downs’s, we now focus on the series itself. To ease the computational burden of PMDL, we choose to only update the parameter estimates every 100 days. We also restrict our attention to the first 6100 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 (41) 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, and the much more elaborate PMDL concurs. This result agrees with our initial work on the up-and-down series from Section 2. Perhaps not surprisingly, this series is much more complex than a simple ARMA process, but our goal here is to illustrate the application of MDL and not dabble in the stock market.

5. Theoretical Results on MDL

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 (cf. 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 the recent review article by Barron et al. (1998).

5.1 Rissanen’s Pointwise Lower Bound

Given a parametric family or model class

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

37
let $E_{\theta}\{ \cdot \}$ denote the expectation with respect to a random variable (data string) $X^n$ having density $f_{\theta}$. (In contrast to previous sections, we are now going to be more careful when referring to random variables $X^n$ versus points $x^n \in \mathbb{R}^n$.) Using this notation, the differential entropy of $f_{\theta}$ defined in (4) 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 $f_{\theta}$ and $q$ is given by

$$R_n(f_{\theta}, q) = E_{\theta}\log \frac{f_{\theta}(X^n)}{q(X^n)} = E_{\theta}\left\{ -\log q(X^n) - \log f_{\theta}(X^n) \right\}. \quad (43)$$

$R_n(f_{\theta}, q)$ represents the expected extra nats needed to encode the data string $X^n$ using $q$ rather than the optimal scheme based on $f_{\theta}$. 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 $M$ reduces to finding a density $q$ that achieves the “smallest” redundancy possible for all members in $M$. To make this concrete, we first derive a lower bound on redundancy in a well-defined global sense over the entire class $M$, and then illustrate choices for $q$ that achieve it. We begin with a pointwise result first derived in Rissanen (1986a).

Assume that a $\sqrt{n}$-consistent estimator $\hat{\theta}(x^n)$ for $\theta$ exists. That is, 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 $q$ and for all $\theta \in \Theta$, Rissanen (1986a) finds that

$$\liminf_{n \to \infty} \frac{E_{\theta}\log \left[ f_{\theta}(X^n)/q(X^n) \right]}{(k/2)\log n} \geq 1, \quad (45)$$

except on a set of Lebesgue measure zero (depending on $q$ and $k$). Viewing $-\log q(X^n)$ as the code length of an idealized prefix code, then (45) implies that without knowing the true distribution $f_{\theta}$, we 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 (Section 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 Section 2). Rissanen’s lower bound (45) extends this result to the case in which we only know that the “true” source belongs to some model class $M$. In coding theory this is referred to as the problem of universal coding. The pointwise lower bound (45) 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 (45) holds for almost every value of $\theta \in \Theta$, hence the term pointwise. We now turn to a minimax version of this result. We again focus 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

$$ R_n^+ = \min_q \sup_{\theta \in \Theta} R_n(f_\theta, q). \quad (46) $$

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 $\theta$. Next, consider a prior distribution $w(\theta)$ on the parameter space $\Theta$ and define the Bayes redundancy associated with a density $q$ relative to $w$ as

$$ R_n^*(q, w) = \int_\Theta R_n(f_\theta, q) w(d\theta). \quad (47) $$

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

$$ R_n(w) = \min_q R_n^*(q, w), \quad (48) $$

which is achieved by the mixture distribution

$$ m^w(x^n) = \int_\Theta f_\theta(x^n) w(d\theta). \quad (49) $$

To see this, write

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

where the last relation holds from Jensen’s inequality. Evaluating (48) at $m^w$ yields

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

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

$$ R_n(w) = I_w(\Theta; X^n). \quad (50) $$

The quantity $I_w$ measures the average amount of information contained in the data $X^n$ about the parameter $\Theta$ and has been used to measure information in a statistical context by Lindley as early as 1956 (cf. Lindley, 1956).

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

$$ R_n^- = \sup_w R_n(w). \quad (51) $$

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 consisting of a value of $\theta$ through a noisy channel represented by the conditional probability of $X^n$ given $\theta$. The receiver
then attempts to reconstruct the message $\theta$ from $X^n$, or rather estimates $\theta$ from $X^n$. Assuming $\theta$ 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 $\theta$, 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 $R^+_n$ (46) from below:

$$R^+_n = \min_{q} \sup_{\Theta} R_n(f_\theta, q)$$

$$\geq \sup_{w} \min_{q} \int_{\Theta} R_n(f_\theta, q) w(d\theta)$$

$$= \sup_{w} \min_{q} R^*_n(q, w)$$

$$= \sup_{w} R_n(w)$$

$$\equiv C(\Theta; X^n),$$

where the equalities (52) and (53) are simply the definitions of the Bayes redundancy (47) and the minimal Bayes redundancy (51), respectively.

Haussler (1995) demonstrates that in fact the minimax redundancy (46) is equal to the channel capacity:

$$R^+_n = C(\Theta; X^n) = R^-_n. \quad (54)$$

According to this result, if we can calculate the capacity of the channel defined by the pair $w$ and $f_\theta$, 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 prior to this point (cf. Le Cam, 1986). See also Davisson (1973), Davisson and Leon-Garcia (1980) and Csiszár (1990).

To be useful, this equivalence requires us to compute the channel capacity for a pair $w$ and $f_\theta$. Unfortunately, this can be a daunting calculation. When both the prior and density function are smooth, however, a familiar expansion can be employed 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 the observation sequence $X^n = X_1, \ldots, X_n$ are iid (or memoryless in the parlance of information theory) from some distribution $f_\theta$ in the class $M$. Under regularity conditions on the prior $w$ and the model class $M$, Clarke and Barron (1990) derived the following expansion in the general $k$-dimensional case (see Ibragimov and Has’minsky, 1973, for the 1-dimensional case). Let $K$ be a compact subset in the interior of $\Theta$. Then, given a positive, continuous prior density $w$ supported on $K$, the Bayes redundancy (47) evaluated at the mixture distribution $m^n$ (49) can be expanded as

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

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, (50)

\[ R_n (w) = I_n (\Theta; X^n) \]

\[ = \frac{k}{2} \log \frac{n}{2\pi e} + \int_{\Theta} w(\theta) \log \frac{\sqrt{\det I(\theta)}}{w(\theta)} \, d\theta + o(1). \]

The middle term is maximized by Jeffreys’ prior (when this prior is well-defined):

\[ w^* (\theta) = \frac{\sqrt{\det I(\theta)}}{\int_{\Theta} \sqrt{\det I(\theta)} \, d\theta}, \]

Hence the minimax redundancy satisfies

\[ R_n^+ = \min_{q \in \Theta} \sup_{\theta} R_n (q, f_{\theta}) = \frac{k}{2} \log \frac{n}{2\pi e} + \log \int_{\Theta} \sqrt{\det I(\theta)} \, d\theta + o(1). \] (55)

Recalling the equivalence (54) and the channel capacity interpretation of the worst case minimal Bayes redundancy, Jeffreys’ 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 \( \theta \) according to Jeffreys’ 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 (55) is the same \( \frac{k}{2} \log n \) as in Rissanen’s pointwise lower bound (45). 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, address qualifying coding schemes based on the constant term). Such codes fairly represent all the members in the model class (in the minimax sense) without the knowledge of exactly which distribution generated our data string.

For more general parameter spaces, Merhav and Feder (1995) prove 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 \( \theta \), hence generalizing Rissanen’s lower bound.

For the case when the source is memoryless, that is, when the observations are conditionally independent given the true parameter \( \theta \), and have a common distribution \( f_{\theta}, \theta \in \Theta \), Haussler and Opper (1995) obtain 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, asymptotic values for minimax redundancy can be obtained 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 \( \frac{k}{2} \log n \) asymptotic lower bounds on redundancy, both in the pointwise and minimax senses. They therefore 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 will now revert to that from Section 4, so that \( x^n \) represents a random sequence \( x_1, \ldots, x_n \).
Example 4 (continued) Two-stage MDL. Trivially, because \( M_0 \) consists of a single distribution, the expected redundancy of \( L_0 \) given in (4) is zero. Now, for \( \theta \neq 0 \),

\[-\log f_\theta(x^n) = \frac{n}{2} \log(2\pi) + \frac{1}{2} \sum_{i=1}^{n} (x_i - \theta)^2.\]

Therefore, the expected redundancy between \( f_\theta \) and the code length function \( L_1 \) (9) is given by

\[E_\theta \{ \log f_\theta(x^n) - L_1(x^n) \} = \frac{n}{2} E_\theta [\bar{x}_n - \theta]^2 + \frac{1}{2} \log n\]

\[= \frac{1}{2} + \frac{1}{2} \log n,\]

which for \( k = 1 \) achieves the pointwise lower bound (45).

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

\[-\log \frac{f_\theta(x^n)}{f_\theta(x^n)}\]

has an asymptotic \( \chi^2_k \) distribution hence its expected value should be \( \frac{k}{2} \), which is of smaller order than \( \frac{1}{2} \log n \). Thus the two-stage description length achieves the lower bound.

Mixture MDL. As with the two-stage scheme, the redundancy of \( L_0 \) is zero because \( M_0 \) consists of a single model. Now, starting with expression (12) we can calculate the expected redundancy for \( L_1 \)

\[\frac{1}{2} (1 + \frac{1}{n\tau}) \log(1 + n\tau) + \frac{1}{2} \times \frac{n}{1 + 1/(n\tau)} E_\theta \bar{x}^2 - \sum_{i} \theta E_\theta x_i + \frac{1}{2} n\theta^2\]

\[= \frac{1}{2} (1 + \frac{1}{n\tau}) \log(1 + n\tau) + \frac{1}{2} \times \frac{n}{1 + 1/(n\tau)} (1/n + \theta^2) - n\theta^2 / 2\]

\[= \frac{1}{2} \log n + O(1),\]

which clearly achieves the pointwise lower bound (45). In addition, given any prior distribution \( w \) on \( \Theta \), we can construct a prefix code according to the mixture distribution \( m^w \) (49). The corresponding code length is

\[L(x^n) = -\log \int w(\theta) f_\theta(x^n).\]

As mentioned above, under certain regularity conditions, Clarke and Barron (1990) showed that the redundancy of the mixture code has the following asymptotic expansion:

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

It follows that the mixture code achieves the minimax lower bound, and as we have mentioned earlier, Jeffreys' prior maximizes the constant term in the minimax redundancy (cf. Barron et al., 1998).
**Predictive MDL.** Using (19), it is easy to check that the redundancy

\[
E_0(- \log q(x^n) + \log f_0(x^n)) = \frac{1}{2} \sum_{t=1}^{n} (1 + 1/t) - n/2
\]

\[
= \frac{1}{2} \sum_{t=1}^{n} 1/t
\]

\[
= \frac{1}{2} \log n + O(1).
\]

Thus it achieves the lower bound (45) and can be used as the description length for data based on model \( M_1 \). As with the previous two forms, the expected redundancy of \( L_0 \) is zero.

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

---

5.4 Assessing MDL 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 (cf. Speed and Yu, 1993). In general, different MDL forms are consistent under very weak conditions (cf. 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.

Example 4 (continued) Recall that two-stage MDL or BIC will select \( M_0 \) if \( |\bar{x}_n| \leq \sqrt{\log n/n} \). When \( M_1 \) is true, the probability of underfitting is

\[
P(M_0 \text{ is selected}) = P_0(|\bar{x}_n| \leq \sqrt{\log n/n})
\]

\[
\approx P_0(N(0,1) \geq 0\sqrt{n} - \sqrt{\log n})
\]

\[
\approx O(e^{-n\theta^2/2}).
\]

Similarly, when \( M_0 \) is true, the probability of overfitting is

\[
P(M_1 \text{ is selected}) = P_0(|\bar{x}_n| > \sqrt{\log n/n})
\]

\[
= P_0(N(0,1) > \sqrt{\log n})
\]

\[
\approx O(1/\sqrt{n}).
\]

Therefore, 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 (cf. Speed and Yu, 1993). Consistency of MDL follows immediately. It also follows from and examination of the underfitting probability that for finite sample sizes, consistency is effected by the magnitude of $\theta^2$ (or squared bias in general) relative to $n$, and not the absolute magnitude of $\theta^2$.

Speed and Yu (1993) 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 where the model bias has a “cliff” or a sharp drop at a certain sub-model class, or when the covariates can be divided into two groups of which one is very important and the other marginal. When model bias decays gradually and never hits zero, however, the consistency criterion does not make sense. In this case, prediction error provides insight into the performance of a selection rule. Shibata (1981) shows 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 in Speed and Yu (1993).

6. Conclusion

In this article, we have reviewed the Principle of Minimum Description length and its various applications to statistical model selection. Through a number of preliminary examples, we have motivated the notion of code length as a measure for evaluating competing descriptions of data. Throughout, our emphasis has been on the practical aspects of MDL. Toward that end, we developed in some detail MDL variable selection criteria in regression problems. As we have seen, the resulting procedures have rich connections with 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.

The descriptive modeling philosophy of MDL grows out of algorithmic complexity theory, and adds to other existing approaches to statistics. We have carefully treated the issue of valid description lengths to make clear the need for justifying a particular code length to be used in any implementation of MDL. This validity issue boils down to the universal coding theorem in information theory and calls for a separate universal coding result to be established in each application. MDL is still an evolving area of active research, and there is no doubt that new forms of MDL will continue to appear. Our understanding of the finite-sample performance of existing MDL criteria will be improved as more and more practitioners apply them to their particular problems. To aid this endeavor, the MDL procedures discussed in this paper will be made available by the first author in the form of an S-PLUS library.
7. Acknowledgements

The authors would like to thank Jianhua Huang for his help with a preliminary draft of this paper. The authors would also like to thank Ed George, Robert Kohn, Wim Sweldens and Martin Wells for helpful discussions.

References


