*DRAFT*

# An Information Theoretic Comparison of Model Selection Criteria

Dean P. Foster & Robert A. Stine

Department of Statistics

The Wharton School of the University of Pennsylvania

Philadelphia PA 19104-6302

February 10, 1997

Abstract

Information theory offers a coherent perspective on model selection. As in Rissanen's original application of information theory to model selection, our perspective arises from viewing a model as a component of a compressed representation of data in a two-part code. The first part of such a code is an explicit representation of the model used to compress the data. Simpler models have shorter representations. The second part is the encoded data itself. Models which fit better compress the data into shorter sequences. The objective is to choose the model which produces the shortest total message length, requiring an explicit trade-off of model complexity (length of the first part) versus goodness-of-fit (length of second part). In addition to Rissanen's *MDL* criterion, this perspective illuminates the properties of numerous model selection criteria, including $AIC$, $C_p$, $BIC$, $RIC$, and $EBIC$. We show that each corresponds to a specific way of coding the model parameters. By selecting the model that minimizes the total message length, our representations of these criteria reproduce their more familiar definitions. Examples from wavelets illustrate the use of these methods.

# 1 Introduction

Given the increasing prevalence of large data sets with numerous predictors, statistical model building faces new challenges. The use of traditional variable selection methods leads to overfitting, characterized by overly complex models that capitalize on chance variation. We attack this problem by using information theory to construct a common framework which encompasses the latest developments in statistical model identification.

All of the currently popular model selection criteria used in regression-type models can be viewed as methods of data compression. From this viewpoint, each criterion is seen to choose the model which minimizes the length of a compressed version of the observed data which we call a 'message'. Each such message is a sequence of bits that unambiguously represents $n$ observed values of a dependent variable $Y_1, Y_2, \ldots, Y_n$. In order to obtain a short message in a regression context, each criterion selects covariates from a collection of potentially useful factors. The better the fit of the model, the shorter the compressed data become. The use of covariates incurs a penalty, however, as the message must identify which covariates have been chosen and describe the associated coefficients. These two tasks distinguish the various criteria: (1) how the message identifies the relevant covariates, and (2) how the message represents the associated coefficients.

The selection criteria discussed here have a long history with a wide range of motivations and behaviours. Each provides an explicit way to capture the principle of parsimony via a penalty for model complexity. Though different in origin, each implies a threshold for including variables in a stepwise regression (Miller 1990). *AIC* originated as a method to minimize the expected Kullback-Leibler distance of the fitted model to the true model (Akaike 1973), and is equivalent in Gaussian regression to $C_p$ (Mallows 1973). *AIC* selects the model which maximizes a penalized log-likelihood, $\log L(\theta_p) - p$, with the penalty term $p$ denoting the number of fitted parameters. In a regression model with orthogonal regressors, *AIC* and $C_p$ select predictors whose $z$-score exceed a threshold of $\sqrt{2}$ in absolute value. A flaw of these criteria is that when presented with data generated by a finite dimensional model, they have a non-vanishing probability of overfitting. Unlike *AIC*, the *SIC*/*BIC* criteria are consistent in this sense. These select the model which maximizes an approximation to the Bayes posterior probability of one of a collection of models being correct (Schwarz 1978, Kass

& Raftery 1995). Here again the criterion chooses a model which maximizes a penalized log-likelihood, but the penalty term is larger and grows with the sample size $n$, $\frac{p}{2}\log n$. Similarly, in the orthogonal regression problem, the threshold for inclusion of a covariate grows to $|z| > \sqrt{\log n}$. The resulting models are more parsimonious than those selected by *AIC*. With an eye toward problems such as wavelet regression with as many (or more) predictors than observations, recent criteria set a threshold based upon the number of predictors $m$ being considered. Hard thresholding (Donoho & Johnstone 1994) and *RIC* (Foster & George 1994) set the threshold for inclusion at $\sqrt{2\log m}$. The resulting model has certain optimal risk properties, predicting as well as a model which fit the right set of predictors (within a term of order $O(\log n)$). Research in multiple comparisons (*e.g.*, Benjamini & Hochberg 1995) implies an adaptive modification of hard thresholding and *RIC*, lowering the barrier to entry as more variables appear useful. Once $p-1$ predictors have been selected, *EBIC* (Foster & George 1996) lowers the threshold for adding the next to $\sqrt{2\log m/p}$.

Though information theory underlies Akaike's motivation for *AIC*, its role in the development of selection criteria is rather limited. An exception, however, is the work of Rissanen (1983, 1989) who explicitly used the notion of coding efficiency for model selection. His minimum description length criterion *MDL* selects the model which is best able to compress the observed data. For an orthogonal Gaussian regression with $p$ covariates chosen from a collection of $m$ predictors, the message length in bits is

$$MDL(p) = m + \frac{p}{2}\log_2 n + \frac{n}{2}\log_2 RSS(p) + q(n)\ , \tag{1}$$

where $RSS(p)$ is the residual sum of squares from the fit and $q(n)$ depends only on sample size and does not affect the comparison. The first two summands in this expression count the number of bits required to identify the chosen covariates and represent the slope estimates. The remaining summands count the number of bits required to encode the data. Thus, *MDL* selects the model which maximizes a penalized likelihood and its penalty term is that of *BIC*.

In what follows, we show that other criteria share a similar characterization but represent the model differently. These differences offer another way to interpret the various regression thresholds. Rissanen's choice of a representation of a regression model leads to a coding method which is equivalent to selecting a model using *BIC*. Other representations

lead to different selection criteria. Information theory offers some appreciation for the relative advantages of these alternatives. Our analysis also suggests methods for developing model selection criteria which are customized for specific problems or are adaptive over a wide range of conditions. Finally, although our characterization suggests some criteria are less desirable than others, we emphasize that each criterion is best within a certain class of problems. For example, the total least squares fit which ignores selection issues obtains the shortest message length when all of the covariates have substantial predictive power.

Following a very brief summary of the needed terminology and results from information theory in §2, we begin our discussion of the coding perspective on model selection with a hypothesis testing problem in §3. Then, in §4, we add covariates to this problem. We offer some illustrative simulations in §5 which contrast the criteria in the selection of coefficients for wavelet reconstructions of simulated data generated by models motivated from stochastic volatility. We close with some concluding remarks in §6.

## 2 Information Theory and Data Compression

Portions of information theory describe how to compress $n$ symbols from some countable alphabet $\mathcal{A}$ into a binary sequence of the shortest expected length. These results underlie file compression tools such as the Unix utility program 'compress'. For our purposes, it is sufficient to consider the case in which the symbols to be sent are realizations of $n$ independent random variables $Y_1, \ldots, Y_n$ taking values in $\mathcal{A}$ with common density $p(y)$. Let $\ell(Y_1, \ldots, Y_n)$ denote the length of the binary message. Then (*e.g.* Cover & Thomas 1991)

$$nH(Y_i) \leq \min E\, \ell(Y_1, \ldots, Y_n) \leq 1 + nH(Y_i) \;,$$

where the entropy $H$ is defined (on a bit scale) as

$$H(Y_i) = -E\{\log_2 p(Y_i)\} = -\sum_{y \in \mathcal{A}} p(y) \log_2 p(y) \;.$$

In the fortuitous case in which the density has the form $p(y) = 2^{-j_y}$ for integers $j_y$, one can see that a code which devotes $j_y$ bits to $Y = y_j$ obtains the entropy bound. The idea is to assign few bits to symbols with high probability, reserving long codes for symbols that are relatively rare. It also implies that the choice of a coding scheme which devotes $k$ bits to

the symbol $a \in \mathcal{A}$ implies that one believes $p(a) = 2^{-k}$; a coding scheme implies a density function and vice-versa.

For other random variables whose density is not of this convenient form, the ability to compress is less obvious but nonetheless remains. For example, suppose the $Y_i$ are Boolean with common probability $\pi = \mathrm{pr}(Y_i = 1) = 1 - \mathrm{pr}(Y_i = 0)$. Then no compression method which treats the symbols separately can compress the data even though $nH(Y_i) < n$ when $\pi \neq 1/2$. Coded separately, each $Y_i$ requires at least one bit, just as in the maximum entropy case with $\pi = 1/2$. In such cases, one can group $Y_1, \ldots, Y_n$ and consider codes for the resulting collection. For example, if $n$ is large and $n\,\pi = 1$, we might consider a code which sends the indices $\{i : Y_i = 1\}$. This can be done by appending a 'continuation bit' onto each index, with a 1 indicating more indices and 0 indicating that the last index has been received. This strategy amounts to sending the number of indices using the geometric (or unary) code $S_g$ shown in the following table:

| $i$ | $S_g(i)$ |
|---|---|
| 0 | 0 |
| 1 | 10 |
| 2 | 110 |
| $\vdots$ | |
| $j$ | $\underbrace{1\ldots1}_{j-1}0$ |

The expected total number of bits required to compress the data in this way is just the length of an index plus the continuation bits. With $n = 2^k$, a Poisson approximation implies that the expected length is about $k + 2$ bits. In the limit, this simple code is slightly longer on average than the optimal code, but comes within a bit of the lower bound:

$$\begin{aligned} \lim_{n\to\infty} n\,H(Y_i) &= \lim_{n\to\infty} n\{-\pi \log_2 \pi - (1-\pi)\log_2(1-\pi)\} \\ &= k + \frac{1}{\log 2} \approx k + 1.44\ . \end{aligned}$$

We term this method of coding using a sequence of indices a 'Poisson code', naming the code after the associated distribution for which it is optimal. In general, algorithms based upon a technique known as arithmetic coding are able to come within one bit of optimal compression

without requiring a special, *ad hoc* analysis of each problem (Bell, Witten, & Cleary 1990). Indeed, given a probabilistic model which assigns probability $\pi_n$ to the observed sequence $y_1, \ldots, y_n$, arithmetic coding can compress this realization into $1 - \log_2 \pi_n$ bits. Thus, given the generating model, the code length of a given realization $y_1, \ldots, y_n$ is within one bit of the log-likelihood (base 2). The problem is thus the choice of this model.

The relevance of message coding for model selection emerges in the Boolean case when $\pi$ is unknown. In this case, the use of compression requires that the sender transmit the value for $\pi$ which was used to encode the compressed bits that complete the message. Without common knowledge of this parameter, the receiver cannot 'undo' the compression and recover the original data. To send this parameter, the sender could begin by choosing to encode the message using the value that offers the most compression of the data at hand, namely the maximum likelihood estimator $\hat{\pi} = \sum Y_i/n$. If $\hat{\pi}$ is close to 1/2, however, the gain from compression (about $n(1 - H(\hat{\pi}))$ bits) can be less than the cost of sending $\hat{\pi}$ (about $\log_2 n$ bits to code $\sum Y_i$). In this case, one obtains a shorter message by sending the raw data instead. The decision of whether or not to code the parameter as part of the message leads naturally to problems in model selection. Further details of this discrete example appear in Rissanen (1989, Examples 5 and 14, p. 57, 117). In the following section we consider the similar case when coding continuous, Gaussian data.

Before moving on, we note that a message comprised of parameters that define the compression algorithm followed by the compressed data is known as a two-part code. These are most natural for model selection. Other methods for compression, such as those like 'compress' which rely upon variants of the Lempel-Ziv algorithm, are one-part codes that are not associated with an explicit statistical model. Although one-part codes typically achieve slightly higher compression than two-part codes (Rissanen 1989, Chapter 3), the absence of an explicit model makes them less useful for statistical model selection.

## 3 Coding and the Gaussian Shift Problem

We now turn to problems in which one is compressing $n$ independent Gaussian observations. Anticipating the coming of fees for the use of the Internet, the goal is to transmit $Y_1, \ldots, Y_n \sim N(\mu, 1)$ to a receiver using as few bits as possible for the message. The first part of the

message identifies an estimate $\hat{\mu}$, and the second part consists of the data compressed using this estimate. To compress such continuous random variables, we assume that the responses have been rounded to $q$ bits to the right of the binary 'decimal' point (as indeed all observed data are rounded). The greater the retained precision, the longer the message becomes. As in the discrete case, the log (base 2) of the likelihood determines the minimum number of bits required to send the data compressed using a given parameter value,

$$\ell(Y_1, \ldots, Y_n; \hat{\mu}) = -\log_2 L(Y_1, \ldots, Y_n; \hat{\mu}) + nq \ ,$$

where $L$ is the likelihood. Clearly, the minimum length of this part of the code is obtained with $\hat{\mu} = \overline{Y}$. Although $\overline{Y}$ gives the best compression of the data, it does not however produce the shortest overall message because sending $\overline{Y}$ precisely makes the first part of the message unnecessarily long. Rissanen (1983, 1989) showed that the minimum overall message length is obtained when $\overline{Y}$ is rounded to order $1/\sqrt{n}$, that is, rounded to a standard error scale. Rounding to this order reduces the length of the first part of the code with a negligible increase on the second part of the code. In general, the increased number of bits required to send $Y_1, \ldots, Y_n$ when the data are coded using some estimator $\hat{\mu}$ rather than $\overline{Y}$ is given by the log likelihood ratio, or relative entropy,

$$\begin{aligned} R(\hat{\mu}, \overline{Y}) &= \ell(Y_1, \ldots, Y_n; \hat{\mu}) - \ell(Y_1, \ldots, Y_n; \overline{Y}) \\ &= \frac{n(\hat{\mu} - \overline{Y})^2}{2 \log 2} \ . \end{aligned} \tag{2}$$

Let $\lfloor z \rceil$ denote the integer nearest $z$ and let

$$z_\mu = \sqrt{n}\, \overline{Y}$$

denote the number of standard errors separating $\overline{Y}$ from zero. If $\overline{Y}$ is rounded to the nearest $j/\sqrt{n}$ as $\tilde{\mu} = \lfloor z_\mu \rceil / \sqrt{n}$, then the data compressed with $\tilde{\mu}$ require less than one bit more than the minimum,

$$R(\tilde{\mu}, \overline{Y}) \leq \frac{1}{8 \log 2} \ . \tag{3}$$

The use of such a code requires an explicit method for representing $\tilde{\mu}$ in the first part of the message. To implement this approach, Rissanen (*e.g.*, 1989, §3.1) places a grid on a bounded parameter space with $|\mu| < M/2$ for some $M > 0$. This approach restricts the

rounded estimator $\tilde{\mu}$ to a grid of $\sqrt{n}\,M$ possible values which can be coded using $\frac{1}{2}\log_2 nM^2$ bits. (Here and generally we ignore the distraction of fractional bits.)

Similar to the example of §2, it is possible to obtain a shorter two-part code by using one bit to represent $\tilde{\mu} = 0$ when $\overline{Y} \approx 0$. The leading bit indicates whether $\tilde{\mu} = 0$, and if not the longer parameter code follows. The lengths of these variations on the first part of the code are (assuming $\sqrt{n}\,M$ is n even integer):

| Parameter Estimate | Length of first part of code |
|---|---|
| $\tilde{\mu} = 0$ | 1 |
| $\tilde{\mu} = \pm j/\sqrt{n}, j = 1, \ldots, \sqrt{n}\,M$ | $1 + \frac{1}{2}\log_2 nM^2$ |

Since the use of a code implies a probability distribution, we call this a 'spike and slab code' because the code assigns probability 1/2 to zero and is uniformly distributed over the remaining grid locations. The parameter is coded as zero whenever $\overline{Y}$ is close enough to zero so that the reduced length of the first part of the code compensates for the increase in the length of the second part of the code. Thus, coding $\tilde{\mu} \neq 0$ requires

$$\frac{1}{2}\log_2 nM^2 < R(0, \overline{Y}) = \frac{z_\mu^2}{2\,\log 2}\ , \tag{4}$$

namely $|z_\mu| > \sqrt{\log nM^2}$. For $M = 1$, this gives the familiar *BIC* threshold.

Graphs defined by the codebook associated with a coding method are particularly useful in understanding the estimator $\tilde{\mu}$ implied by a particular two-part code. A codebook describes how the parameter is encoded in the first part of the message and is in effect a dictionary of (value, binary string) pairs. More formally, a codebook $C$ is a one-to-one relation $S$ defined on a countable set $\mathcal{T}$ which maps each member of $\mathcal{T}$ into a binary string, $S : \mathcal{T} \to \{0,1\}^j$, $j = 1, 2, \ldots$. The set $\mathcal{T}$ defines a grid on the parameter space $\Theta$; these are the only values for $\tilde{\mu}$ which can be encoded in the message. The map $S$ determines the sequence of bits which identify the parameter. For the previous encoding of Rissanen, the codebook $C_b$ consists of the grid (again treating $\sqrt{n}\,M$ as an even integer)

$$\mathcal{T}_b = \{M/2, \ldots, -1/\sqrt{n}, 0, 1/\sqrt{n}, \ldots, M/2\}$$

with

$$S_b(t) = \begin{cases} 0, & t = 0, \\ B_k\{\iota(t, \mathcal{T}_b - \{0\})\} & t \neq 0\ , \end{cases}\ ,$$

where $k = \frac{1}{2}\log_2 nM^2$, $B_n(j)$ is the $n$ bit binary representation of the integer $j$, and $\iota(t, T)$ is the index of $t$ in the ordered set $T$. Given a codebook $C = \{\mathcal{T}, S\}$, the encoder selects for the estimator the grid location $t \in \mathcal{T}$ yielding the shortest message. In the Gaussian case, the total message length is

$$Q_C(\overline{Y}, \tilde{\mu}) + \log_2 L(y_1, \ldots, y_n; \overline{Y}) + nq \ , \tag{5}$$

where

$$Q_C(\overline{Y}, \tilde{\mu}) = \ell\{S(\tilde{\mu})\} + R(\hat{\mu}, \overline{Y}) \ , \quad \tilde{\mu} \in \mathcal{T} \ , \tag{6}$$

and $\ell(b)$ denotes the length of the binary string $b$. The quadratic $Q_C(\overline{Y}, \tilde{\mu})$ gives the increase in total message length caused by using the estimator $\tilde{\mu}$. Hence, the encoder first chooses $\tilde{\mu}$ to be the value minimizing the varying component of the message length,

$$\tilde{\mu}_C = \arg\min_{t \in \mathcal{T}} Q_C(\overline{Y}; t) \ . \tag{7}$$

The encoder then sends $S(\hat{\mu}_C)$ as the first part of the code and next uses $\hat{\mu}_C$ to compress the data as the second part of the code. In order to decode the message, the receiver must know the codebook used by the encoder. Given the codebook, the receiver first inverts the encoding relation to recover $\hat{\mu}_C$ which is then used to decode the compressed data.

Graphs of quadratics $Q_C(\overline{Y}; \tilde{\mu})$ are particularly useful. Figure 1 shows $Q_{C_b}(\overline{Y}; j/\sqrt{n})$ for $j = 0, \ldots, 10$ of the *BIC* codebook with $n = 1024$ and $M = 4$. The horizontal axis of the figure is scaled to show the standardized mean $z_\mu$, thereby centering the quadratics at integers. The quadratic $Q_{C_b}(\overline{Y}; 0)$ centered at zero is of particular interest since its relationship to the others determines whether a nonzero parameter is coded. It indicates the contribution to the message length caused by coding $\tilde{\mu} = 0$ as a function of $\overline{Y}$. If indeed $\overline{Y} = 0$, this code contributes one bit to the total message length. As $\overline{Y}$ moves away from zero, the impact of ignoring $\overline{Y}$ increases quadratically due to the deteriorating data compression. The quadratics centered at other integer $z$-scores show the lengths obtained by coding $\tilde{\mu} = j/\sqrt{n}$. The figure also highlights in bold the function $\min_{t \in \mathcal{T}} Q_{C_b}(\overline{Y}; t)$ which traces the bottoms of the minimizing quadratics. Note that four members of $C_b$ are shadowed by the quadratic centered at zero and would never be used. (The two which are visible in the figure are dashed.)

A different parameter code produces a criterion which resembles *AIC* with a fixed threshold below two. Again, assume that $\overline{Y}$ is rounded to the grid $j/\sqrt{n}$. Rather than use a fixed number of bits for a nonzero parameter, drop the assumption that $|\mu| < M/2$ and instead encode the rounded values using the 'Cauchy code' $S_c(k)$ illustrated in the following table.

| $k$ | $k_2$ | $S_c(k)$ | Bits with Sign |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 1 | **1**0+ | 3 |
| 2 | 10 | **11**0+ | 5 |
| 3 | 11 | **11**1+ | 5 |
| 4 | 100 | **110**10+ | 7 |
| 8 | 1000 | **1101010**0+ | 9 |

This code interleaves the binary representation of an integer with a sequence of continuation bits, with the final zero bit indicating the end of the sequence. A sign bit (shown as the symbol '+' in the table) follows codes for nonzero integers. The codebook is then $C_a = (\{j/\sqrt{n} : j \in \mathbf{Z}\}, S_c(j))$ with $\mathbf{Z}$ denoting the integers. In later asymptotic calculations we use the approximation $\ell\{S_c(j)\} \approx 1 + 2\log_2 j$. In effect, the leading bit of the Cauchy code acts as the zero/nonzero choice bit of the previous code. Figure 2 graphs the codebook $C_a$, again for $n = 1024$. As seen in the figure, one codes a non-zero value for $\tilde{\mu}$ once $\overline{Y}$ is about 2 standard errors above zero. Specifically, one starts to code $\tilde{\mu} = 1$ once

$$3 + R(1, \overline{Y}) < 1 + R(0, \overline{Y}) \,, \tag{8}$$

which occurs at $z_\mu \approx \pm 1.89$. Since the encoded estimator $\tilde{\mu} = 1/\sqrt{n}$ at this point, the minimization has introduced a slight amount of shrinkage. This type of parameter coding resembles the *AIC*, coding a non-zero parameter as part of the model once $|\overline{Y}|$ is a fixed distance from zero. In this case, the distance is slightly larger than the usual *AIC* threshold of $\sqrt{2}$. Unlike the behaviour of the uniform code, the threshold does not increase with $n$. As with the previous code, some members of this codebook are never used; these are the quadratics associated with $\sqrt{n}\,\tilde{\mu} = 2^j$, $j = 1, 2, \ldots$ and are highlighted by dashed curves in Figure 2.

Whereas the previous code lost a few long codebook members, the effect of the shadowing for this code is more important since some of the unused parameterizations would make the first part of the code very short. Figure 2 makes it evident that one can improve this code by simply moving the shadowed codebook members farther from zero. Referring to Figure 2, we obtain a uniformly shorter code by sliding the quadratics centered at $\pm 1$ out to about $\pm 1.665$, moving those farther away from zero out by a corresponding amount. This small change also moves the threshold to code a non-zero parameter to its minimum value $|z_\mu| > 1.665$ and closer to the *AIC* threshold. Continuing, the code is improved at each member by shifting the quadratic to the right enough so that its minimum is 'exposed.' A particularly simple way to obtain this effect is to change the rounding scheme. For example, round $z_\mu$ to a more coarse grid so that the codebook consists of the pair $(\{2j/\sqrt{n} : \ j \in \mathbf{Z}\}, \ S_c(j))$. The graph of this codebook appears in Figure 3. The threshold for coding with two standard error spacing is about $|z_\mu| \approx 1.69$, almost the minimum possible. Figure 4 compares the minimum increments to the message lengths obtained by rounding to 1, 2 or 3 standard errors with the minimum obtained from the *BIC* codebook $C_b$ with $M = 4$, chosen arbitrarily to separate the code lengths in the figure. The codebooks based on various Cauchy codes yield shorter messages than the previous spike-and-slab code when $\overline{Y}$ is near zero — values near the usual null hypothesis. As $|\overline{Y}|$ increases, the spike and slab code eventually produces shorter messages as long as $|\overline{Y}| < M/2$.

Some terminology is useful for describing codebooks. A *perturbation* $P(C)$ *of a codebook* $C = (\mathcal{T}, S)$ is a modification of one of the members of the defining grid set $\mathcal{T}$, shifting an associated quadratic. A perturbation of a codebook is a *dominating perturbation* if when coding any sequence $Y_1, \ldots, Y_n$ in a two part code, the perturbed codebook produces a message of shorter length than the original codebook, $\ell_{P(C)}(Y_1, \ldots, Y_n) < \ell_C(Y_1, \ldots, Y_n)$. In the one-parameter Gaussian shift situation, this condition is equivalent to $\ell_{P(C)}(\overline{Y}) < \ell_C(\overline{Y})$ for all $\overline{Y}$. Finally, a codebook is *dominance stable* if there exists no dominating perturbation. A codebook is dominance stable if for all $t \in \mathcal{T}$, $Q_C(t;t) = \min_{s\in\mathcal{T}} Q_C(t;s)$. In this case, $\tilde{\mu} = \overline{Y}$ whenever $\overline{Y} \in \mathcal{T}$. Dominance stable codebooks do not show the degree of shrinkage obtained, for example, with the codebook $C_a$. Graphically, a codebook is dominance stable if the minimum of each quadratic is exposed. From Figure 3, the codebook based on rounding

to two standard errors produces uniformly shorter messages than $C_a$ and is dominance stable. The codebook using three standard error spacing is dominance stable, but does not lead to uniformly shorter messages. No simple change to the grid locations of either produces a uniformly shorter message length.

To summarize this section, the following table contrasts these two approaches to hypothesis testing via coding in the mean shift problem:

| Attribute | Codebook | |
|---|---|---|
| | $C_b$ | $C_a$ |
| Parameter code | Spike and slab | Cauchy |
| Parameter space | $[-M/2, M/2]$ | $\mathbf{R}$ |
| Selection criterion | *BIC* | *AIC* |
| Parameter threshold | $\lvert z_\mu \rvert > \sqrt{\log nM^2}$ | $\lvert z_\mu \rvert > 1.89$ |

## 4 Regression Model Selection

The coding methods introduced in the Gaussian shift problem generalize to multiparameter problems and lead to direct characterizations of various model selection criteria in regression. Each of *AIC*, *BIC*, *RIC*, and *EBIC* corresponds to a specific way of identifying the relevant covariates and representing the fitted parameters in the first part of a two-part code. These codes corresponding to each criterion represent the parameters of the model differently, and so reach different compromises of model complexity and goodness of fit.

To develop our comparison, we consider the problem of variable selection from a collection of $m$ potential orthogonal covariates in a regression with Gaussian errors having known variance $\sigma^2 = 1$. Our focus is upon problems with large numbers of covariates, here limited by orthogonality to $m \leq n$ as in wavelet regression. In keeping with the previous section, we adopt the convention that each covariate is normalized so that $\|X_j\|^2 = n$ and denote the least squares estimates associated with a set of $p$ covariates as $\hat{b}(p) = (\hat{b}_{j_1}, \ldots, \hat{b}_{j_p})$. As when fitting a mean, a variable improves the model if the number of bits required to add its rounded coefficient to the first part of the code is smaller than the gain in data compression obtained in the second part of the code (so that the overall message length is decreased).

As in §3, the improvement in data compression offered by adding a variable is proportional to the change in the log-likelihood. The log-likelihood based on $p$ predictors is (ignoring constants)

$$-\log L = \tfrac{1}{2}\|Y - \hat{Y}(p)\|^2 = \tfrac{1}{2}(\sum Y_i^2 - \sum z_j^2) \ ,$$

where the fitted values are $\hat{Y}(p) = \boldsymbol{X}_p\hat{b}(p)$, $\boldsymbol{X}_p = (X_{j_1}, \ldots, X_{j_p})$ and the $z$-score for $X_j$ is $z_j = \sqrt{n}\,\hat{\beta}_j$. Adding another predictor, say $X_k$, to the fit reduces the residual sum of squares by $z_k^2$, implying that the compressed data require $z_k^2/(2\log 2)$ fewer bits. We denote the rounded $z$-score by $\lfloor z_j \rfloor$ and the rounded estimator by $\tilde{\beta}_j = \lfloor z_j \rfloor/\sqrt{n}$. Once again, the degradation in data compression from rounding this coefficient is less than one bit as in (3), $R(\tilde{\beta}_j, \hat{b}_j) \le 1/(8\log 2)$, though in aggregate the cost for rounding can be substantial. If $\tilde{Y}(p) = \boldsymbol{X}_p\tilde{\beta}(p)$, then the overall loss of compression from rounding is proportional to

$$\|Y - \tilde{Y}(p)\|^2 - \|Y - \hat{Y}(p)\|^2 = \|\boldsymbol{X}_p\{\hat{b}(p) - \tilde{\beta}(p)\}\|^2 = n\|\hat{b}(p) - \tilde{\beta}(p)\|^2 \le \frac{p}{4} \ .$$

The two previous codes associated with *BIC* and *AIC* adapt easily to orthogonal regression. The spike-and-slab method introduced in the shift problem represents the regression model with two components: a prefix of $m$ bits $a_1, \ldots, a_m$ where $a_j = 1$ implies $X_j$ is in the fit, followed by codes for the parameters. Each parameter is represented by coding the rounded z-score $\lfloor z_j \rfloor$ using $\frac{1}{2}\log_2 n^2M$ bits (assuming the bound $|\beta_j| < M/2$). As in the location problem leading to (4), the threshold for adding a parameter is the *BIC* rule $|z_j| > \sqrt{\log nM^2}$. The previous Cauchy code also adapts easily. The explicit $m$ bit prefix used in the *BIC* format is absorbed by the Cauchy codes for the parameters. That is, $a_1, \ldots, a_m$ are the first bits of each Cauchy parameter code. For example, if $p = 7$ and $\lfloor z_j \rfloor = (0, 1, 0, 3, 0, 0, 2)$, then the model parameters are coded as

$$\underline{0} \quad \underline{1}\,0 \quad \underline{0} \quad \underline{1}\,1\,1\,0 \quad \underline{0} \quad \underline{0} \quad \underline{1}\,1\,0\,0$$

The seven underlined bits are $a_1, \ldots, a_7$. (The spaces separating the codes are visually useful but unnecessary since the Cauchy code is self-delimiting.) This coding implies that the change in the likelihood and consequent improved data compression is enough to overcome the addition of a parameter to the first part of the code when $\hat{b}_j$ is approximately 2 standard errors away from zero, $|z_j| > 1.89$, as in §3. Again, one can move this threshold by changing

the rounding grid. This ability to vary the threshold is analogous, and helps interpret, the effect of changing the penalty from $p$ to, say, $2p$ in variations on *AIC* such as those considered in Bhansali & Downham (1977).

Several more recent model selection criteria may also be characterized as two-part codes. Hard thresholding (Donoho and Johnstone 1994) and the risk inflation criterion *RIC* (Foster and George 1994) include a variable in the model whenever its coefficient satisfies $|z_j| > \sqrt{2\log m}$. These criteria are equivalent to choosing the model that minimizes the following code. Rather than use a prefix of $m$ bits to identify the $p$ included covariates, this code uses a Poisson code and describes the fitted model using $p$ pairs of the form $(j, \lfloor z_j \rfloor)$ with the index $j$ coded uniformly in $\log_2 m$ bits and $\lfloor z_j \rfloor$ represented by a Cauchy code. The number of pairs $p$ is indicated with a geometric code; that is, a continuation bit is added to each pair to indicate if more pairs follow. With the index explicitly paired with the coefficient, this code extracts a higher penalty for adding a variable. Comparison to the gain in data compression implies that one adds a variable whenever

$$\frac{z_j^2}{2\,\log 2} > 1 + \log_2 m + S_c(\lfloor z_j \rfloor)\ , \tag{9}$$

implying the desired asymptotic threshold for large $m$ of $|z_j| > \sqrt{2\log m}$. Figure 5 shows a graph of the codebook $C_r$ based upon this message format with $m = n = 1024$. For example, the height of the quadratic $Q_{C_r}(1/\sqrt{n}, \hat{b})$ at its center $1/\sqrt{n}$ is 14: one for the continuation bit, three for $S_c(1)$, and 10 bits for the index of the coefficient.

One can also create a codebook which leads to the type of estimators and threshold given by soft thresholding. Soft thresholding retains the threshold $\sqrt{2\log m}$ implied by *RIC*, but shrinks nonzero parameter estimates by this amount so that the fitted $z$ score is $\hat{z}_j = |z_j - \sqrt{2\log m}|^+$ (for positive $z_j$). As noted in §2, shrinkage requires the use of a codebook which is not dominance stable, so the code shown here is more illustrative than efficient. The parameters are coded using a signed version of the geometric code $S_g$ introduced in §2, so that 0 is coded with one bit, 1 with three bits, 2 with four bits, and so forth. Given that the code for the first nonzero parameter is 2 bits longer than that for zero, the needed shrinkage implies that the first nonzero parameter that is coded is located at approximately $(2\log 2)/\sqrt{2\log m}$ for large $m$. Subsequent grid values are spaced at half this

distance since the relative parameter lengths differ by one bit. When $m = n$ as in the wavelet simulations in §5, this grid is a finer partition of the parameter space than the optimal (in the sense of message length) $1/\sqrt{n}$ spacing used by the other codes discussed here. Figure 6 graphs $Q(\hat{b}, 0)$ and $Q(\hat{b}, (j+1)\log 2/\sqrt{2\log\, m})$, ($j$=1,. . . ,4, and $j = 5, 10, \ldots, 40$) from a soft thresholding codebook with $m = 1024$. The exposed right sections of the shown quadratics produce an approximately linear function that shrinks $\hat{b}$ to the respective centers highlighed by the small dots in the figure. The small vertical line shows the threshold $\sqrt{2\log m}$.

Foster and George (1996) have offered a modification of *RIC* called *EBIC* which is motivated through empirical Bayes. Their methods are related to multiple comparison procedures discussed in Benjamini and Hochberg (1995). This selection criterion is adaptive, with a lower barrier to inclusion once several covariates have been included in the fitted model. Explicitly, the criterion implies that the coefficient for $X_j$ is added to a model with $p$ coefficients once

$$|z_j| > \sqrt{2\log\{m/(p+1)\}} \tag{10}$$

An adaptive two-part code gives an asymptotically equivalent criterion. The $m$ bit prefix $a_1, \ldots, a_m$ included with the spike and slab code or as the first bits of the Cauchy codes is an efficient code (in the sense of being short) only when this sequence of indicators behaves as a sequence of independent Bernoulli random variables with probability $\mathrm{pr}(a_j = 1) = \frac{1}{2}$. That is, this method produces short codes for the indicators only when about half of the predictors can be expected to be useful. In problems with many covariates, one often expects few valuable predictors and the resulting code for the $m$ indicators is wasteful. At the other extreme, the Poisson code implicit in hard thresholding and *RIC* implies a much smaller probability for inclusion, $\mathrm{pr}(a_j = 1) = 1/m$. If indeed the number of predictors is Poisson with mean one, then this indexing method is efficient. Either code implicitly assumes *a priori* an expected number of predictors. Alternatively, one can compress the indicator bits efficiently without such an assumption by sending a special prefix consisting of the number of predictors $p = \sum a_j$ and a code that identifies $\{j : a_j = 1\}$. This prefix requires only $\log_2 m + \log_2 \binom{m}{p}$ bits and is within about $(\log_2 m)/2$ of the entropy coding limit since

$$\log_2 \binom{m}{p} = mH(p/m) - \tfrac{1}{2}\log_2 m - \tfrac{1}{2}\log_2(1 - p/m) - \log_2 e + O(1)\ ,$$

where the remainder is less than one bit. This message format compresses the selection of the predictors within a few bits of the compression achieved by a code which is given $p$. To see that this code reproduces the *EBIC* criterion, the increase in data compression is greater than the increase in the length of the coded model whenever (assuming $p < m/2$ so that $\log_2 \binom{m}{p}$ is increasing in $p$)

$$\frac{z_j^2}{2\,\log 2} > \log_2 \frac{m-p}{p+1} + \ell\{S_c(\lfloor z_j \rfloor)\} \;,$$

or once $|z_j| \approx \sqrt{2 \log m/p}$, approximately for $m >> p$. Figure 7 graphs the associated codebook $C_e$ for several values of $p/m$. When $p/m = 1/1024$ (Figure 7a), the codebook resembles that in Figure 5 for the *RIC* code $C_r$. Coding a single 1 as part of the compression of $a_1, \ldots, a_{1024}$ costs 10 bits, enough to give the index of the single nonzero coefficient. At $p/m = 1/2$, this method behaves just like the *AIC* code in Figure 2 since the cost for encoding each $a_j$ is one bit. Finally, once most of the variables have been included in the fit, say $p/m > 0.9$ (Figure 7b), this procedure will necessarily include the rest of the coefficients with a nonzero estimate, even if the least squares estimator $\hat{b} = 0$. This choice occurs because the cost of indicating some $a_j = 0$ as part of the compression of $a_1, \ldots, a_{1024}$ exceeds the cost of coding the slope $\hat{b}$ as one. If $\hat{b}_j = 0$, then it costs $-\log_2(1 - p/m)$ to code $a_j = 0$, indicating $\tilde{\beta}_j = 0$. In contrast, the number of bits to code $a_j = 1$ and $\tilde{\beta}_j = \pm 1$ is

$$-\log_2 p/m + 2 + R(1/\sqrt{n}, 0) \;,$$

where $2 = S_c(1) - 1$ since the leading bit of the Cauchy code for 1 is included in the compressed indicator sequence $a_1, \ldots, a_m$. When $p/m > 1/(1 + 2^{-2.72}) \approx 0.868$, the length for coding $\tilde{\beta}_j = 0$ is longer than the length for coding $\tilde{\beta}_j = \pm 1$ even when the least squares estimator $\hat{b}_j = 0$. At this point, the *EBIC* threshold is zero and all of the remaining coefficients are included in the fit.

## 5 Simulation

Models of stochastic volatility produce data that exhibit the well-known tendency of financial time series to show trends in variation. For such series, the variation in one time period can

be used to predict the variation in nearby periods. A simplified model of this tendency is represented by the diffusions

$$dX(t) = e^{\theta(t)} dW_1(t), \qquad d\theta(t) = \lambda dW_2(t) \ , \tag{11}$$

where $W_1(t)$ and $W_2(t)$ are two possibly correlated Brownian motions. In applications, one observes a discrete time series formed by sampling $X(t)$ and seeks to recover and predict the instantaneous standard deviation $e^{\theta(t)}$. Further discussion of this model and related ARCH models appears in the collection of Rossi (1996). Estimation for this model brings complications outside our interest here, but the model suggests an important class of problems in which the signal of interest is a very irregular function.

Our simulations consider a stylized version of this problem. The data are simulated from the classical white noise model used frequently in the study of nonparametric regression,

$$X_t = f(t) + \sigma \epsilon_t \ , \qquad 0 \le t \le 1 \ ,$$

where $\epsilon_t \sim N(0,1)$, independently. Unlike many applications in which the signal $f(t)$ is smooth, for our simulations $f(t)$ is a scaled Brownian bridge,

$$f(t) = \sqrt{6}\{W(t) - tW(1)\} \ , \qquad 0 \le t \le 1 \ ,$$

with the scale factor chosen so that $E\{\sum_{i=1}^{n} f(i/n)^2\} \approx n$. The Brownian bridge has the roughness suggested by $\theta(t)$ in (11) as well as the periodicity of the basis functions used in our wavelet regressions. The data series in the simulation have varying signal to noise ratios $\eta^2$, obtained varying the signal strength while fixing $\sigma = 1$. Each data series $\boldsymbol{X}_j = (X_{j1}, \ldots, X_{jn})'$ consists of $n = 1024$ observations formed as

$$X_{jt} = \eta \, f_j(t/1024) + \epsilon_t, \qquad t = 1, \ldots, 1024 \ , \tag{12}$$

where the Brownian bridge $f_j$ is simulated independently of $\epsilon_t$ for each realization. In the sense of regression, the best $R^2$ for a fit is $R^2 = \eta^2/(1+\eta^2)$. Given $\boldsymbol{X}_j$, $\eta f$ is estimated from the wavelet coefficients computed as

$$c_j = W_4 \boldsymbol{X}_j \ ,$$

where $W_4$ is the $n \times n$ orthogonal basis matrix associated with the standard periodic Daubechies wavelet $d_4$ (*e.g.*, Donoho & Johnstone 1994). Each of the four selection methods was then applied to the fit $c_j$, using the fact that the standard error of $c_j = 1$. For the *BIC* scheme, we set $M = 1$ so that the threshold implicit in our implementation is consistent with the usual definition (1), though our interpretation suggests a larger value.

The graphs in Figures 8, 9, and 10 summarize the mean squared error of the reconstructions for 4000 simulated series with the square root of the signal to noise ratio ranging over the interval $0 \leq \eta \leq 50$. Figure 8 shows the mean squared error of 2000 wavelet reconstructions based on coding implementations of *AIC*, *BIC*, *RIC*, and *EBIC*. Of the 2000 simulated realizations of (12), 1000 series are uniformly distributed over $0 \leq \eta \leq 50$, with an additional 1000 in the smaller interval $0 \leq \eta \leq 10$. Figure 9 shows the relative mean squared error of these same reconstructions, first over the full range of $\eta$'s, and then focussing on $\eta \leq 10$. The relative mean squared error compares the fit of these reconstructions to that obtained by an optimal threshold. The optimal threshold for each realization is found numerically by choosing that threshold which minimizes the mean squared error for the given realization, as though an oracle had give the data analyst the proper threshold. Figure 9 also includes the reconstruction based on the data itself; that is, a reconstuction using the complete least squares fit with effective threshold zero. The horizontal line in Figure 8 at $y = 1$ summarizes the least squares fit in this case since $\sigma = 1$ throughout. In the two figures, a lowess curve summarizes the accuracy of each estimator. Because this smoother conceals the variation about the trend and is not reliable near the boundary at zero, we have provided some additional figures which compare the selection criteria at fixed values of $\eta$. Figure 10 highlights the differences among the estimators at four smaller values $\eta = 0.25$, 1, 2, and 4 (or, $R^2 =$ 0.06, 0.5, 0.67, and 0.80). The frames of this figure show comparison boxplots of the relative mean squared error at the given signal to noise ratio. Each uses a separate simulation of 500 realizations. To help untangle these summaries, Figure 11 shows the proportion of fitted nonzero coefficients which were coded by the *AIC* and *EBIC* procedures, plotted on $\eta$. The *AIC* codebook fits more coefficients until $\eta \approx 35$, at which point the adaptive behaviour of the *EBIC* codebook begins to code more coefficients. For large $\eta > 40$, this version of *EBIC* codes all of the coefficients as nonzero values.

When $\eta$ is small, few coefficients rise above the noise floor. In this situation, *RIC* and *EBIC* perform the best. Although this is a small part of the range covered in the figures, it can be argued that this is the most important region to perform well since for a large class of functions, most of the wavelet coefficients are near zero. Thus, fitting well in this portion of the figure may be most important for most applications. Because of its bias toward few coefficients, the mean squared error of *RIC* reconstructions quickly rises as $\eta$ increases. In contrast, because it is more adaptive, *EBIC* reconstrcutions remain competitive until $\eta \approx 4$. Once $\eta \approx 4$, the liberal coding of *AIC* becomes effective and it chooses models with the smallest mean squared error until $\eta \approx 35$. (For smaller values of $\eta$, *AIC* performs quite poorly as seen in Figure 9). With such strong signals, the adaptive coding of the *EBIC* procedure begins to fit more parameters and it once again obtains relatively better performance. Note that the plots of the relative mean squared error occasionally show that *AIC* gives a model with smaller mean squared error than one which knows the optimal threshold. This seeming contradiction is explained by noting that the optimal threshold estimator does no rounding or shrinkage of the estimated coefficients.

## 6 Discussion

We have not addressed here the matter of how to handle models with correlated parameters. Such an extension is important for practical regression problems and specialized applications like smoothing splines. It would also allow us to handle non-nested subset problems in, for example, time series model selection. Nested models, such as the common comparison of consecutive order autoregressions, can be addressed with our current results via a simple orthogonalization implied by the nesting. Our assumption of orthogonality implies an additivity to the code lengths, making graphs of the univariate codebooks apply in regression as well.

A second extension of this work is the use of these alternative model selection strategies in data compression algorithms. This use requires that we modify them to work with adaptive Markov models known as context trees. Some of the most successful data compression algorithms compress a sequence of bits $Y_i$ using an arithmetic coder which is supplied conditional probabilities of the form $\mathrm{pr}\{Y_t = 1 | Y_{t-1}, \ldots, Y_{t-s_t}\}$ from a statistical model. The

conditioning information, here $Y_{t-1}, \ldots, Y_{t-s_t}$, is known as a context, and the size $s_t$ of the context varies. Bell, Witten and Cleary (1990) provide examples of this algorithm and find it among the best. The important aspect regarding model selection is the question of how to "prune" an initial $2^m$ binary tree using the compression ideas described above. Commonly, a codebook resembling the *BIC* codebook $C_b$ is used. In some preliminary experiments, we have had some success compressing binary sequences using alternative coding methods, such as Cauchy parameter codes enhanced with variance stabilizing transformations. A natural generalization of the binary context tree is to the problem of "dynamic" autoregressive models, replacing the sequence of bits by continuous random variables. The structure of the context trees also suggests the opportunity to develop analogous model selection methods for the various partitioning algorithms used in statistics, particularly CART (Breiman, Friedman, Olshen, Stone 1984) and MARS (Friedman 1991) which recursively divide the prediction space into subsets in which a homogeneous model is fit.

Finally, although the paradigm of model selection based on code length can be quite powerful and lead to important heuristics, it leaves us with an important question: What are the statistical properties of the resulting estimators? For example, it is well-known that estimators implied by the *MDL* criterion are consistent when a "true model" is known to generate the observed data. We appreciate that many readers may not be persuaded by simply "counting bits" when choosing a model, and consequently plan to explore the risk properties of the estimators implied by these models. Although the simulation evidence of §5 is perhaps compelling for some, we clearly need to address the risk properties implied by the estimators associated with each codebook and establish the linkage between the length of a binary message and risk.

# 7 References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory*, B. N. Petrov & F. Csàki, Eds., Akad. Kiàdo, Budapest, 261-81.

Bell, T. C., Cleary,J. G., and Witten, I. H. (1990). *Text Compression*. Prentice-Hall, Englewood Cliffs NJ.

Benjamini, Y. & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. of the Royal Statist. Soc., Ser. B*, **57**, 289-300.

Bhansali, R. & Downham, D. Y. (1977). Some properties of the order of an autoregressive model selected by a generalization of Akaike's FPE criterion. *Biometrika*, **64**, 547-51.

Breiman, L., Friedman, J. H., Olshen, R. and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont CA.

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

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaption by wavelet shrinkage. *Biometrika*, **81**, 425-455.

Foster, D. P. & George, E. I. (1994). The Risk Inflation Criterion for multiple regression. *Annals of Statistics*, **22**, 1-41.

— (1996). Empirical Bayes variable selection. In preparation.

Friedman, J. H. (1991). Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, **19**, 1-141.

Kass, R. E. & Raftery, A. E. (1995). Bayes factors. *J. Amer. Statist. Assoc.*, **90**, 773-95.

Mallows, C. L. (1973). Some comments on $C_p$. *Technometrics*, **15**, 661-75.

Miller, A. J. (1990). *Subset Selection in Regression.* London, Chapman & Hall.

Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, **11**, 416-431.

— (1989). *Stochastic Complexity in Statistical Inquiry*, World Scientific, Singapore.

Rossi, P. (1996). *Modelling Stock Market Volatility.* San Diego, Academic Press.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6**, 461-4.

**Figure 1.** Graph of the *BIC* equivalent codebook $C_b$ for $n = 1024$ with the parameter space restricted to $|\mu| < 2$.

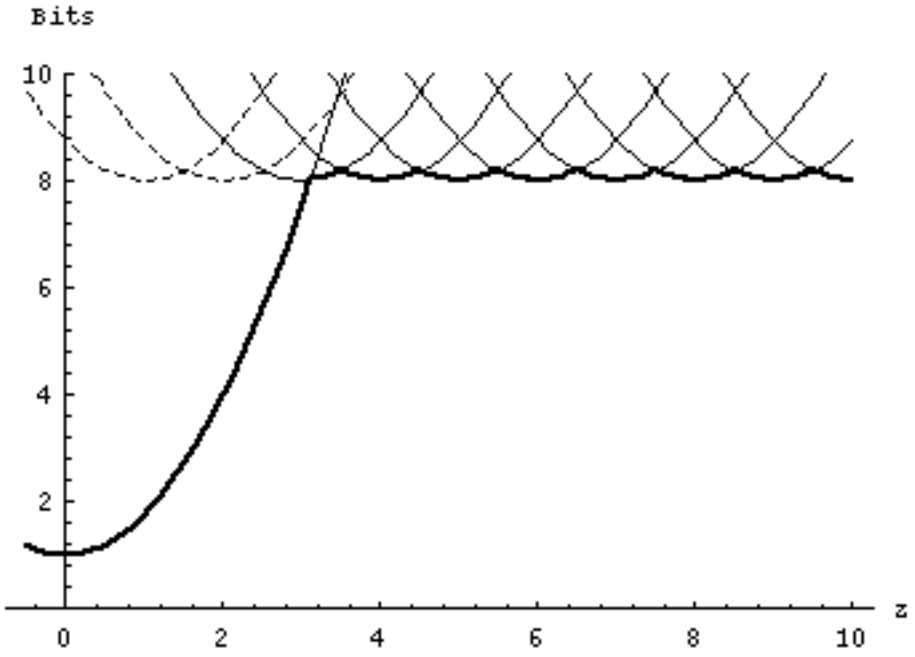


**Figure 2.** Graph of the $AIC$ equivalent codebook $C_a$ for $n = 1024$ with one standard error spacing.

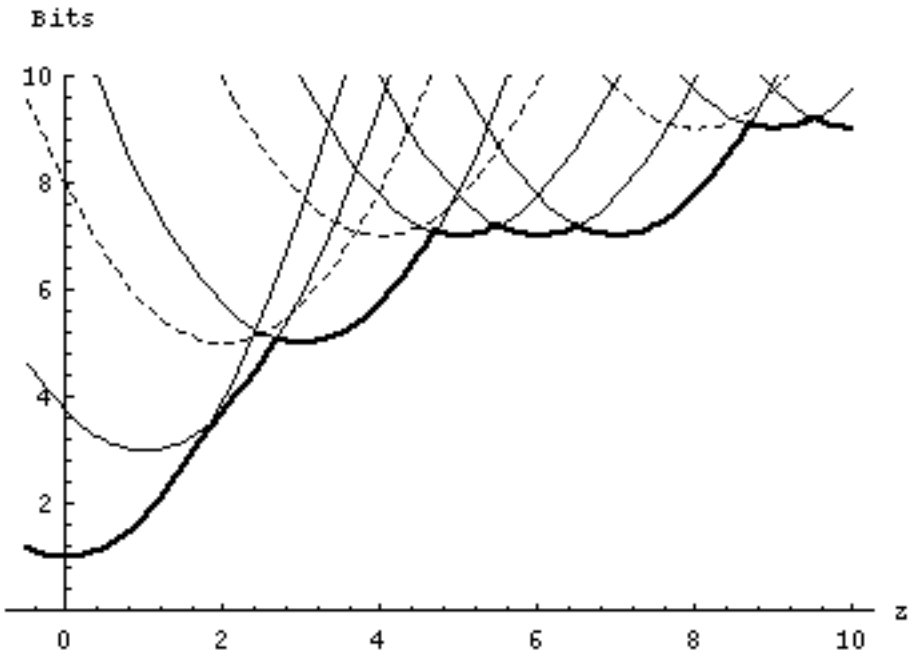


**Figure 3.** Graph of the *AIC* equivalent codebook $C_a$ for $n = 1024$ with two standard error spacing.

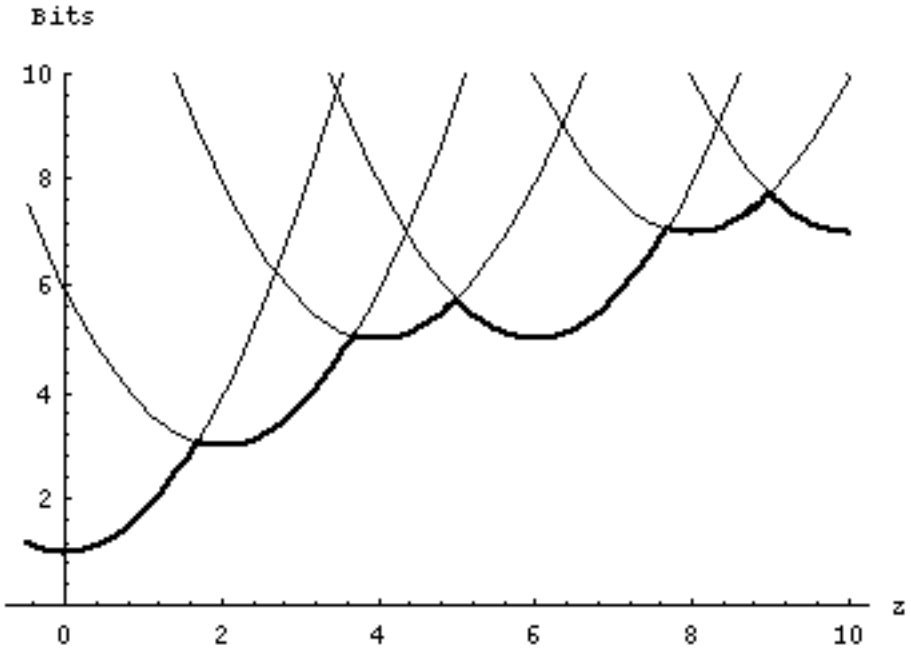


**Figure 4.** Comparison of the minimum additional message bits required by the *BIC* codebook $C_b$ and the *AIC* codebook $C_a$ with varying standard errors ($n = 1024$ and $M = 4$).

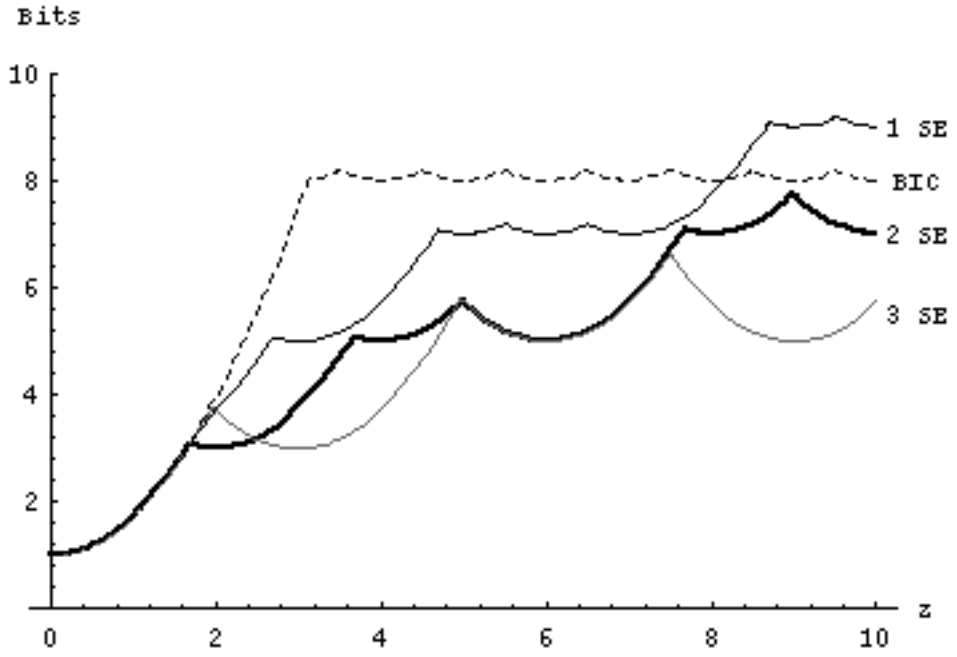


**Figure 5.** Graph of the *RIC* equivalent codebook $C_r$ for $n = 1024$ with one standard error spacing.

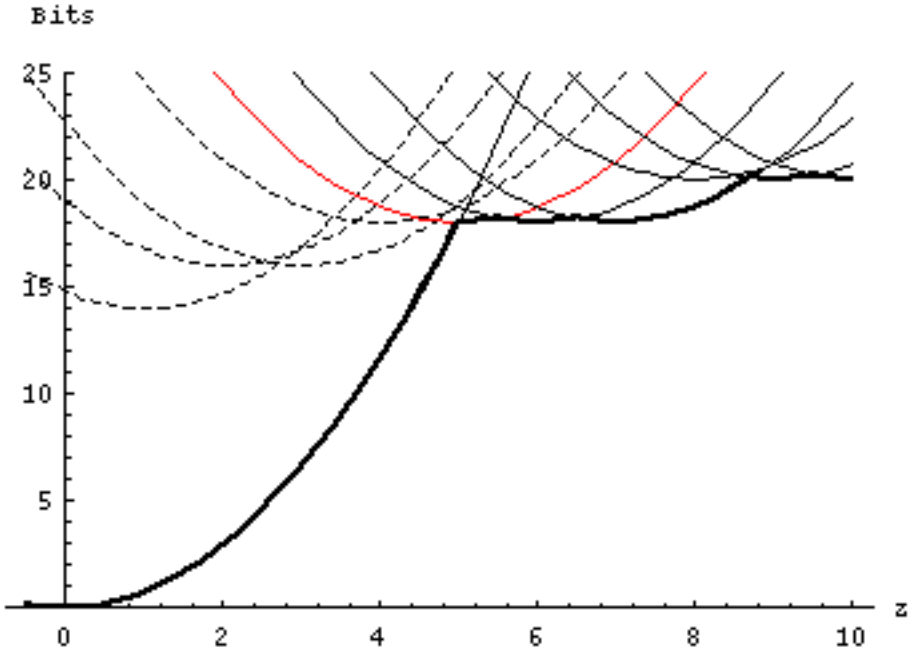


**Figure 6.** Graphs of $Q(\hat{b}, 0)$ and $Q(\hat{b}, (j{+}1) \log 2/\sqrt{2 \log p})$, ($j$=1,. . . ,4, and $j = 5, 10, \ldots, 40$) from a soft thresholding codebook with $p = 1024$. The small vertical line at $x \approx 3.7$ indicates the threshold $\sqrt{2 \log p}$.

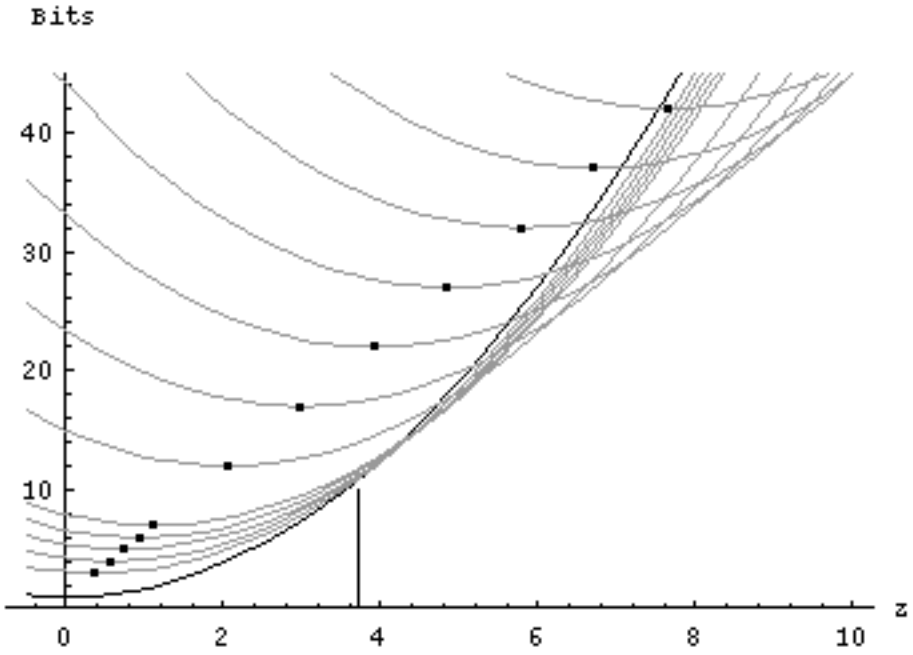
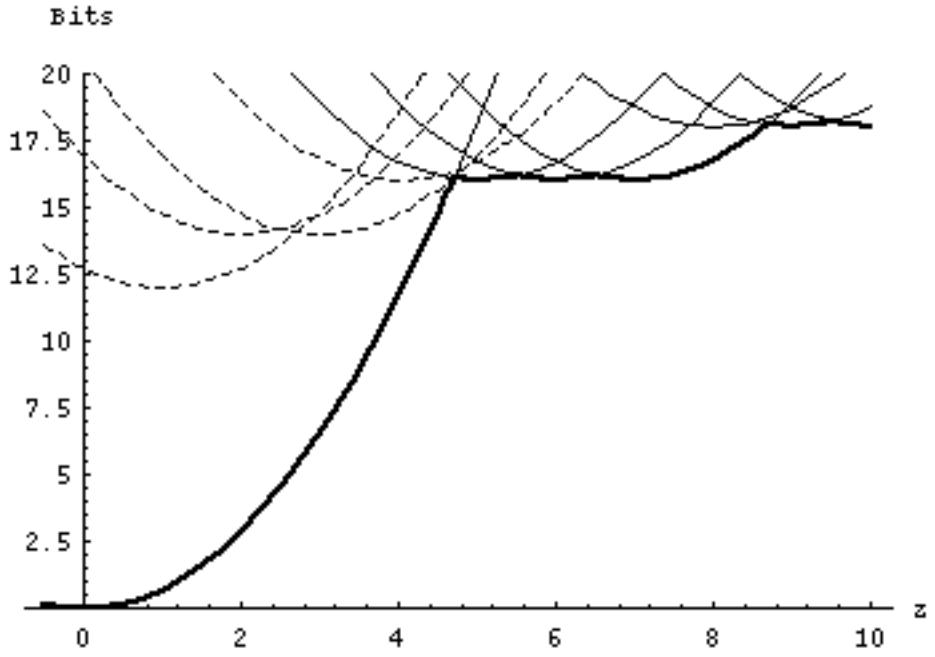


**Figure 7.** Graph of the *EBIC* equivalent codebook $C_e$ for $n = 1024$ with one standard error spacing. (a) With $p/m = 1/1024$. (b) With $p/m = 0.90$.

(a)




(b)

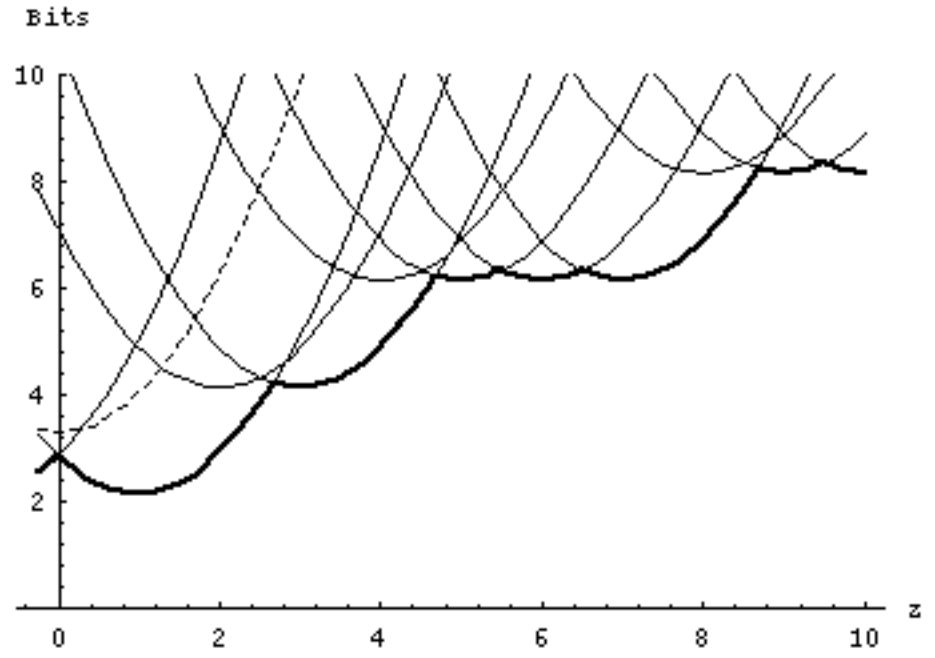


**Figure 8.** Graph of the ratio of th mean squared error of wavelet reconstructions for square root of the signal to noise ratio $0 \leq \eta \leq 50$ ($AIC$ $\circ$, $BIC$ $+$, $RIC$ $\times$, $EBIC$ $\mathsf{2}$).

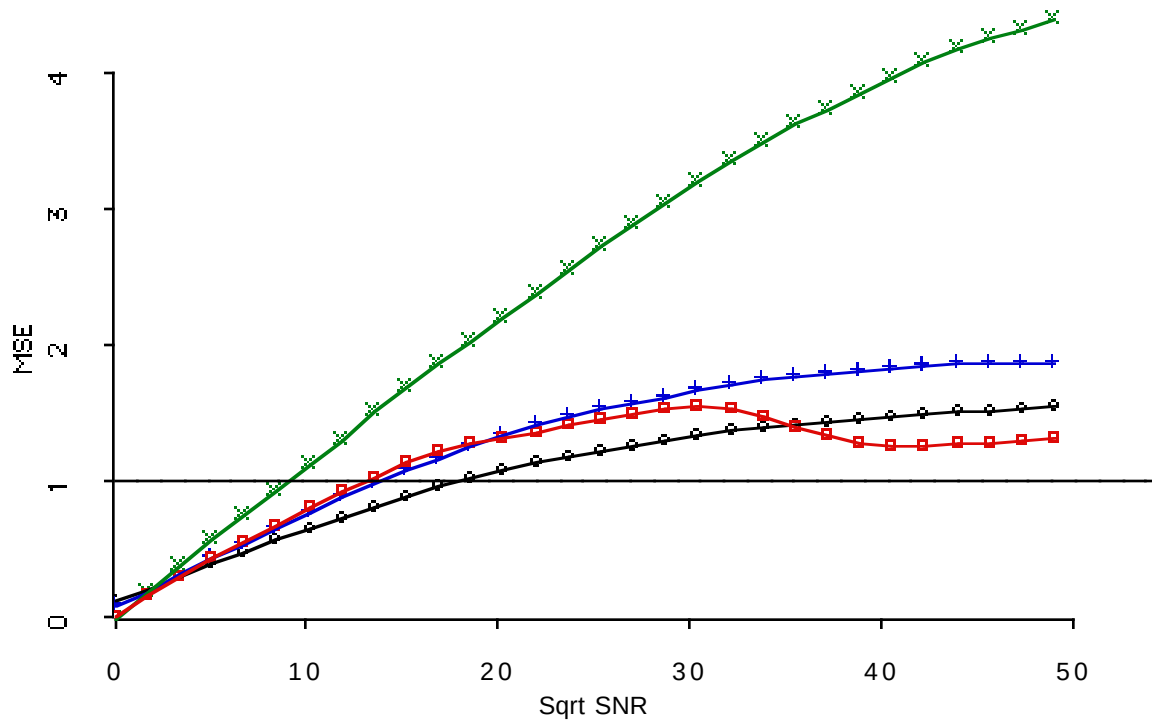


**Figure 9.** Graph of the ratio of the relative mean squared error of wavelet reconstructions for signal to noise ratio (a) $0 \leq \eta \leq 50$ (b) $0 \leq \eta \leq 10$ ($AIC$ $\circ$, $BIC$ $+$, $RIC$ $\times$, $EBIC$ $\mathbf{2}$, data/full least squares $\cdot$).

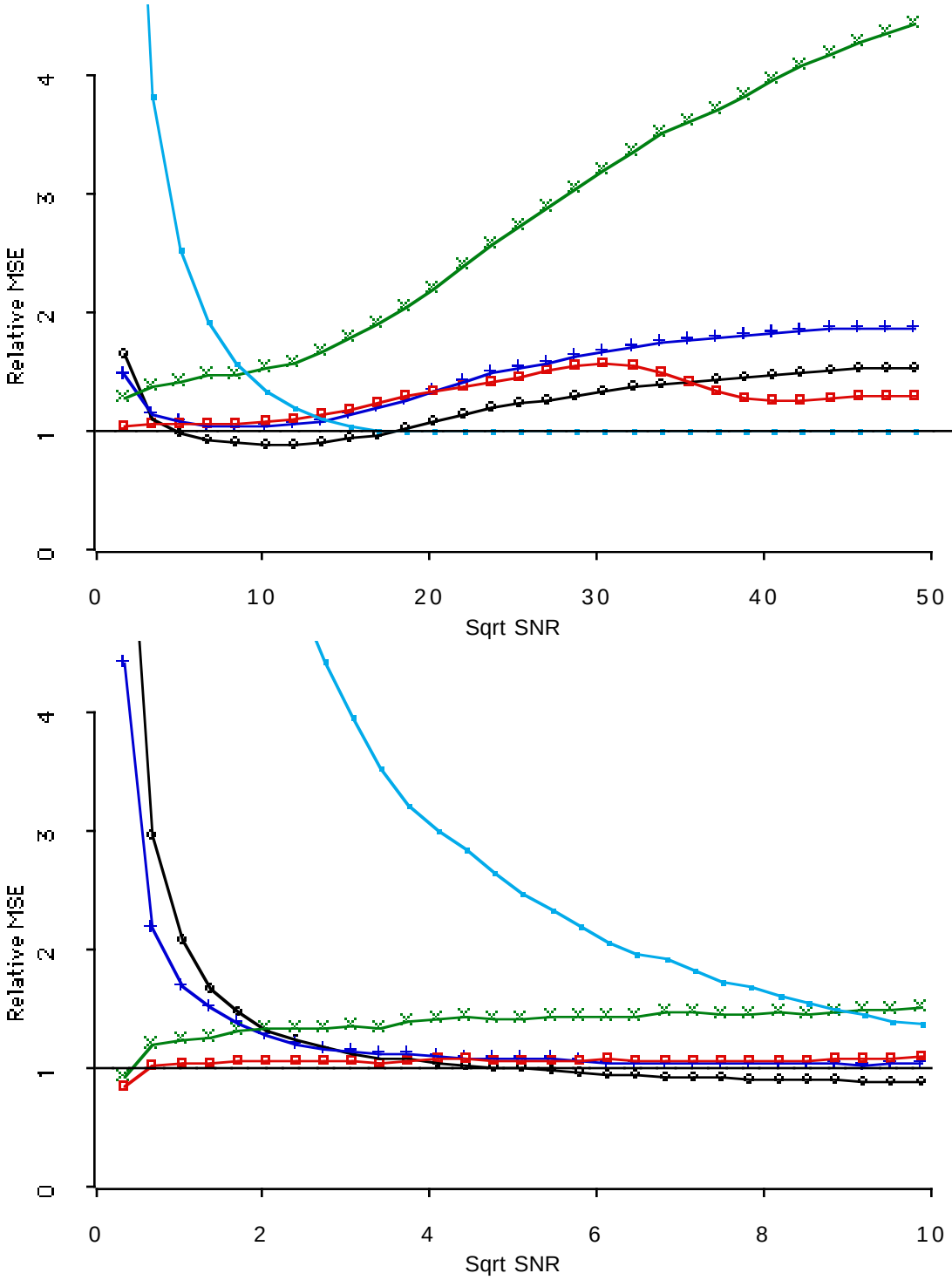
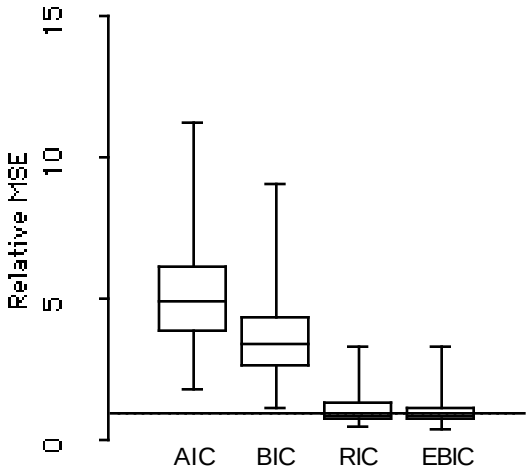
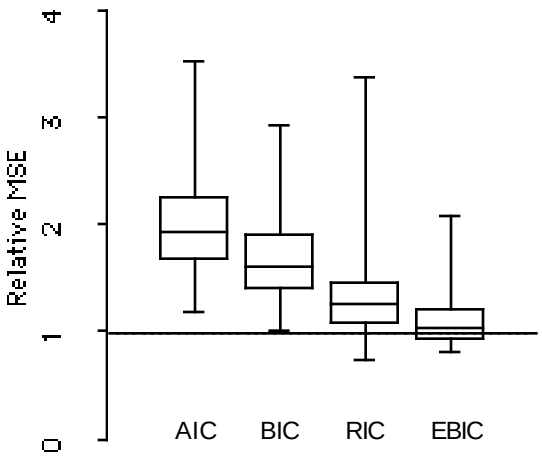


**Figure 10.** Comparison boxplots of the relative mean squared error of wavelet reconstructions using *AIC*, *BIC*, *RIC*, and *EBIC*for varying values of the signal to noise ratio $\eta = 0.25$, 1, 2, 4.

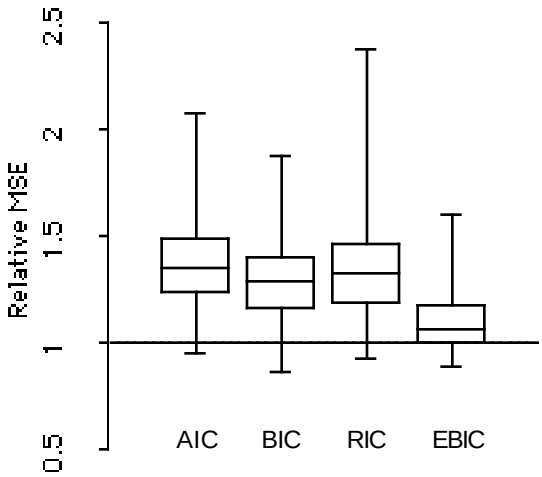
(a) $\eta = 0.25$




(b) $\eta = 1$




(c) $\eta = 2$




(d) $\eta = 4$

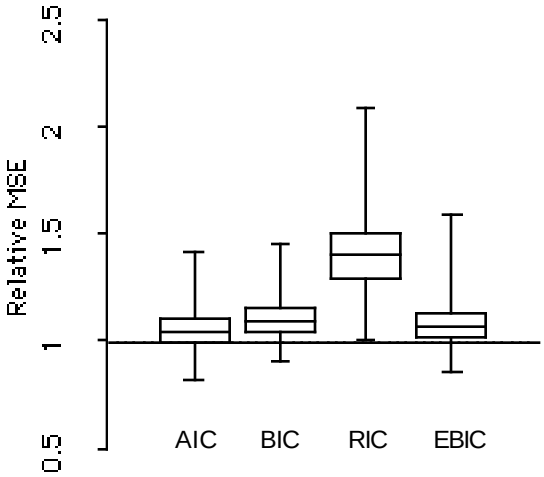


**Figure 11.** Plots of the number of fitted nonzero coefficients as coded by the *AIC* ( ∘ ) and *EBIC* ( 2 ) procedures. The shown coordinates are for a sample of 200 of the simulated 1000 replications in each case.

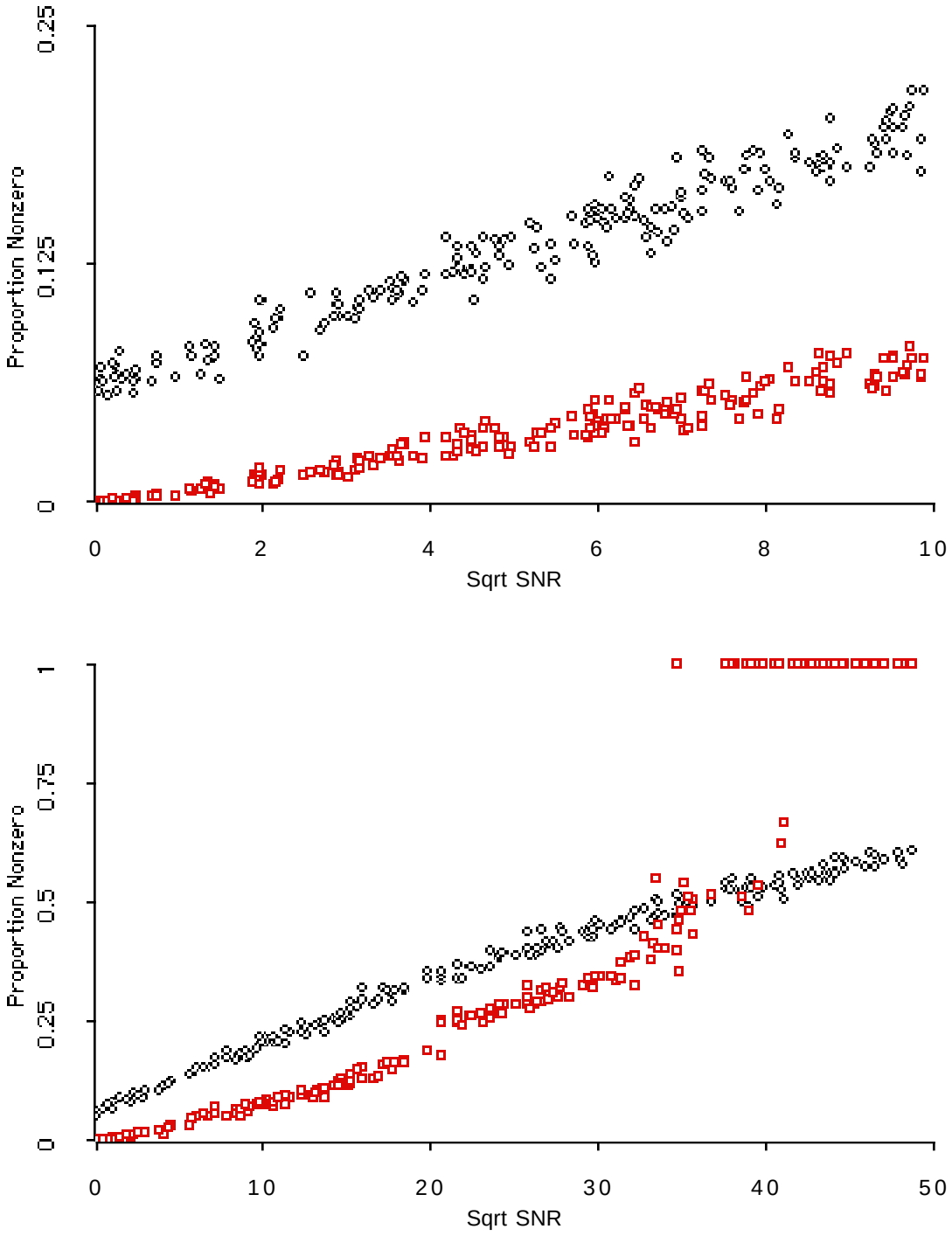


**Figure XX.** Comparison boxplots of the relative mean squared error of wavelet reconstructions using *AIC*, *BIC*, *RIC*, and *EBIC*for varying values of the signal to noise ratio $\eta = 10, 40$.

(a) $\eta = 10$ (b) $\eta = 40$