

## *Coding Heuristics for Model Selection*

### Overview

1. MDL, model selection, and coding
2. The location problem
3. Testimators and selection
4. Applications to least squares problems
5. Several codes: uniform, Cauchy, indexed, adaptive
6. Discussion

### MDL, model selection, and coding

**Representative problem** Which variables ought to be used in a regression or ARIMA model, particularly when the number of predictors is as large or larger than number of observations  $p \geq n$ .

**Model selection = data compression** Model selection via most popular criteria (*AIC*, *BIC*, *RIC*) is equivalent to choosing model that which offers the shortest lossless description of the data.

Criterion	Threshold	Origin
Least squares	0	Gauss
<i>AIC</i>	$\sqrt{2}$	Expect divg (Akaike 1973)
$C_p$	$\sqrt{2}$	Prediction (Mallows 1973)
<i>BIC, SIC</i>	$\sqrt{\log n}$	Bayesian (Schwarz 1978)
<i>MDL</i>	$\sqrt{\log n}$	Compression (Rissanen 1983)
<i>RIC</i>	$\sqrt{2 \log p}$	Risk (Foster & George 1994)
hard thresh	$\sqrt{2 \log p}$	Minimax (Don & Johns 1994)
<i>EBIC</i>	$\sqrt{2 \log q/p}$	Emp. Bayes (F & G 1996)

**Two-part codes** The compressed data are represented by a two-part code,  
Two-part code: Model + Data

The selection criteria differ in how they encode the model as part of this compressed representation.

**Information theory** Perspective as data compression offers

- Consistent perspective for the various criteria.
- Comparison of criteria.
- Suggests alternative criteria, customized criteria.

**Key terms**

- Entropy: Expected value of the negative log-likelihood. For the r.v.  $X$

$$H(X) = \mathbb{E} \log 1/p(x) \tag{1}$$

Entropy determines minimum expected message length (discrete),

$$H(X) = - \sum_x p(x) \log_2 p(x) \Rightarrow \min_{\ell} E \sum \ell(X_i) = nH(X)$$

Optimal obtained (within one bit) using a code with lengths

$$\ell(x) = -\log_2 p(x)$$

- Relative entropy, divergence: Expected value of the likelihood ratio implied by one probability distribution  $Q$  relative to the true distribution  $P$ ,

$$D(P \parallel Q) = \mathbb{E}_P \log P(X)/Q(X) \geq 0 \tag{2}$$

which has the interpretation as the expected excess code length when data  $X \sim P$  is encoded by  $Q$ .

**Implications**

- High compression devotes short codes to likely symbols.
- Optimal code synonymous with pdf,  $p(x) = 2^{-\ell(x)}$ .

**MDL** Rissanen's criterion for picking a model is to choose the model (or probability distribution) that has the minimum message length (or minimum description length). The description length for data  $X$  obtained by the model  $P_\theta$  is (this is the two-part length)

$$L_2(X, P_\theta) = \underbrace{\log P_\theta(X)}_{\text{log likelihood}} + \underbrace{L(\theta)}_{\text{code for } \theta} \tag{3}$$

## The location problem

**Task** Send  $Y_1, \dots, Y_n \sim N(\mu, 1)$  to a receiver using as few bits as possible, but allow the receiver to recover the data without loss.

**Quantization** How do you encode real-valued data, such as  $Y \sim N(\mu, \sigma^2)$ ? Round (*i.e.*, ‘quantize’) to some accuracy. Number of bits required to code  $y_1, \dots, y_n$  is

$$\log_2 L(y_1, \dots, y_n; \theta) + nQ$$

where level of precision determines  $Q$ .

**Notation** Given  $X_1, \dots, X_n \sim p(x; \theta)$ , then likelihood is

$$L(\theta) = p(x_1; \theta) p(x_2; \theta) \cdots p(x_n; \theta)$$

Let  $\hat{\theta}$  denote the MLE.

**Trade-off** Optimal compression for  $Y$ ’s obtained using  $\hat{\mu} = \bar{Y}$ , but how can the receiver decode the message? Must also include  $\hat{\mu}$  as part of the two-part code. Ideas in this regard are:

1. Don’t send  $\hat{\mu}$  to full precision. Rounding to  $O(1/\sqrt{n})$  saves bits sending parameter and causes small increase in length of compressed data. Send  $\hat{\mu}$  rounded to standard error scale (*z-score*) using, say,  $\frac{1}{2} \log_2 n$  bits.
2. Don’t include the parameter if its near zero, just send zero instead (flag costs one bit). Two types of messages:

0 + raw quantized data

vs

1 +  $z_{\hat{\mu}}$  + data compressed with  $\tilde{\mu}$

**Parameter rounding** Negligible effect. Effect of coding with a rounded parameter are on the order of one bit in coding  $n$  observations. For example, if data  $X_1, \dots, X_n \sim N(\mu, 1)$ , and if code using MLE

$$X_1, \dots, X_n \Rightarrow -n \log_2 L(\bar{X}) + nQ \text{ bits}$$

If use rounded parameter  $\tilde{\mu}$  s.t.

$$|\tilde{\mu} - \bar{X}| < \frac{1}{2\sqrt{n}},$$

then length used for the data increases by about

$$\log_2 L(\tilde{\mu}) - \log_2 L(\bar{X}) = n \log_2 e(\bar{X} - \tilde{\mu})^2 < 1 \text{ bit}$$

but we save much more for the smaller parameter.

## Testimators and selection

**Testimator** Declare that the parameter  $\mu$  is zero if the message

0 + raw quantized data

is shorter than

1 +  $z_{\hat{\mu}}$  + data compressed with  $\tilde{\mu}$

### *BIC* penalty

Coding the parameter as part of the message costs  $\frac{1}{2} \log_2 n$  bits. With  $n = 256$ , prefer to code  $\tilde{\mu}$  (*i.e.*, estimate a nonzero parameter) once  $z \approx 4.5$ , and this threshold increases logarithmically with  $n$ .

### Alternative criteria

To arrive at other criteria, change how the  $z$ -score is coded in the first part of the message. Otherwise, stay the same:

- Still code a rounded parameter ( $z$ -score).
- Length of data remains log-likelihood.

### Idea

Vary the number of bits used to code the  $z$ -score. Rather than reserve  $\frac{1}{2} \log_2 n$  bits for rounded  $\hat{\mu}$ , use few bits when close to zero (the “null” value) and more as it gets farther away (“local coding”).

### Cauchy code

Interleave binary representation of integer  $z$ -score with 1’s as continuation codes and zero as a stop bit. The length in bits is about  $\ell_c(z) \approx 2 \log_2 z$ , so optimal if  $p(z) = 2^{-\ell_c(z)} \approx z^{-2}$ .

$z$	Binary $z$	Code for $z$	$\ell_c(z)$
0	0	0	1
1	1	10	2+1
2	10	1100	4+1
3	11	1110	4+1
4	100	110100	6+1
8	1000	11010100	8+1

For example,

$$\underbrace{0}_0 \underbrace{10}_1 \underbrace{0}_0 \underbrace{1110}_3 \underbrace{0}_0 \underbrace{0}_0 \underbrace{1100}_2$$

**AIC-like selection** The special “flag bit” which denoted whether  $\tilde{\mu}$  was part of message no longer appears; it’s part of the Cauchy code. With this code and  $(n = 256, \sigma^2 = 1)$ , one obtains shorter messages “AIC like”  $z \approx 2$ , invariant of  $n$ .

**Inefficient** Some codes are not used. Optimal spacing of  $z$  values that are coded?

## Applications in Gaussian least squares

### Trade-off

Add a variable when resulting data compression “pays for” increase in complexity in first part of the code.

### Model

Use  $q$  of  $p$  predictors  $Y = \beta_0 + \beta_{j_1} X_{j_1} + \dots + \beta_{j_q} X_{j_q} + \epsilon$  where  $X'X = nI_q$  and  $Z_j = \sqrt{n}(\hat{\beta}_j - \beta_j) \stackrel{\text{iid}}{\sim} N(0, 1)$ .

### Gain in compression

Log-likelihood based on  $q$  predictors in a least squares fit (Gaussian likelihood) is (ignoring constants)

$$-\log L = \frac{1}{2} \sum_{i=1}^n (Y_i - \hat{Y}_i(q))^2 = \frac{RSS(q)}{2}$$

If add another predictor, say  $X_m$ , then

$$RSS(q + 1) = RSS(q) - n \hat{\beta}_m^2 = RSS(q) - Z_m^2$$

so compressed data require

$$\frac{Z_m^2}{2 \log 2} \text{ fewer bits.}$$

**Parameter cost**

Least squares, *BIC*, *AIC*, *RIC*, and *EBIC* code the parameters differently, and so reach different compromises of data compression and model complexity.

**Uniform coding**

**First part of code**

Use  $p$  bits  $b_1, \dots, b_p$  to indicate which variable is included, then code in  $\frac{1}{2} \log_2 n$  bits for those included,  $\{j : b_j = 1\}$ .

**Second part of code**

Compressed data as in least squares.

**Two-part code**

$$b_1, \dots, b_p \mid \underbrace{\text{uniform codes for } \langle Z_{j_1} \rangle, \dots, \langle Z_{j_q} \rangle}_{\frac{q}{2} \log_2 n} \parallel \underbrace{\text{compressed data}}_{-\log_2 L+nQ}$$

**Resulting selection**

Add  $X_m$  if the goodness of fit compensates for adding  $\frac{1}{2} \log_2 n$  bits for the additional parameter,

$$\underbrace{\frac{Z_m^2}{2 \log 2}}_{\text{Increased compression}} > \underbrace{\frac{1}{2} \log_2 n}_{\text{Increased param bits}}$$

implying that one codes if (recall assumption that  $|\beta_j| < M = \frac{1}{2}$ )

$$|Z_m| > \log n .$$

## Cauchy coding

### First part of code

Eliminate the explicit  $p$  bit prefix and simply concatenate the Cauchy codes for  $\langle Z_j \rangle$ :

$$\mathbf{0 . 1 0 . 0 . 1 1 1 0 . 0 . 0 . 1 1 0 0} \Rightarrow \mathbf{0 1 0 3 0 0 2}$$

Leading characters of Cauchy code *are* the p-bit prefix.

**Second part of code** Compressed data as in least squares.

### Two-part code

$$\underbrace{\text{cauchy codes for } \langle Z_1 \rangle, \dots, \langle Z_p \rangle}_{\sum_{j=1}^p \ell_c \langle Z_j \rangle} \parallel \underbrace{\text{compressed data}}_{-\log_2 L+nQ}$$

### Resulting selection

Add  $X_m$  if improved goodness of fit compensates for adding  $\ell_c(\langle Z_m \rangle)$  bits for the additional parameter,

$$\underbrace{\frac{Z_m^2}{2 \log 2}}_{\text{Increased compression}} > \underbrace{\ell_c(\langle Z_m \rangle) \approx 2 \log_2 \langle Z_m \rangle}_{\text{Increased param bits}}$$

which implies, as before, that one codes once

$$|Z_m| > 2$$

## Indexed coding

### First part of code

Fewer bits indicate which coefficients are being used. Denote coefficients by (index,value) pairs,

$$(j, \langle Z_j \rangle)$$

$\langle Z_j \rangle$  denoted by a Cauchy code.

**Second part of code** Compressed data as in least squares.

### Two-part code

$$q \mid \underbrace{(j_1, \langle Z_{j_1} \rangle)}_{\log_2 p + \ell_c(\langle Z_{j_1} \rangle)} \mid \dots \mid \underbrace{(j_1, \langle Z_{j_q} \rangle)}_{\log_2 p + \ell_c(\langle Z_{j_q} \rangle)} \parallel \underbrace{\text{compressed data}}_{-\log_2 L+nQ}$$

**Resulting selection**

Add  $X_m$  if (approximately)

$$\underbrace{\frac{Z_m^2}{2 \log 2}}_{\text{Increased compression}} > \underbrace{\log_2 p + 2 \log_2 \langle Z_m \rangle}_{\text{Increased param bits}}$$

or roughly once

$$|Z_m| > \sqrt{2 \log p}$$

**Adaptive coding**

**First part of code**

Adaptive, efficient code for  $p$  bit selection prefix. (Recall  $p \geq n \gg q$ ).

Method	Parameter Coding	Bit Length
<i>BIC</i>	Explicit prefix	$p$
<i>AIC</i>	Implicit prefix	$p$
<i>RIC</i>	Indexing	$q \log_2 p$
<i>EBIC</i>	Compressed code	$p H(q/p) \approx \log_2 q + \log_2 \binom{p}{q}$

**Second part of code** Compressed data as in least squares.

**Two-part code**

$$\underbrace{b_1, \dots, b_p}_{p H(q/p)} \mid \underbrace{\text{'Cauchy codes' for } \langle Z_{j_1} \rangle, \dots, \langle Z_{j_q} \rangle}_{\sum_{k=1}^q \ell_c \langle Z_{j_k} \rangle - q} \parallel \underbrace{\text{compressed data}}_{-\log_2 L + nQ}$$

**Resulting selection**

Add  $X_m$  if (approximately)

$$\frac{Z_m^2}{2 \log 2} > p \left( H\left(\frac{q+1}{p}\right) - H\left(\frac{q}{p}\right) \right) + 2 \log_2 \langle Z_m \rangle$$

or once

$$\frac{Z_m^2}{2} > \log \frac{p-q}{q+1} + 2 \log Z_m \Rightarrow |Z_j| > \sqrt{2 \log q/p}$$

## Discussion

### Model selection message formats

Criterion	Threshold	Parameter	Selection Prefix
<i>BIC, SIC</i>	$\sqrt{\log n}$	Uniform	$p$ bit explicit
<i>AIC, C<sub>p</sub></i>	$\sqrt{2}$	Cauchy	embedded
<i>RIC, hard</i>	$\sqrt{2 \log p}$	Cauchy	index (Poisson)
<i>EBIC</i>	$\sqrt{2 \log q/p}$	Cauchy	compressed

**One-part codes** Two-part codes are not “tight” in the sense that they reserve codewords that will never be used. One part codes resolve this problem but lose the intuitive connection to model selection and parameterized models. In a one-part code, it becomes difficult to separate the data from the parameters; there’s not a clean break between the two. We lose the “penalized likelihood” metaphor.