Akaike’s Information Criterion (AIC)

- Based on:
  - the maximization of the expected log-likelihood.
  - The fact that the maximum log-likelihood is a biased estimator of the expected log-likelihood, with a bias equal to the number $k$ of free parameters in the model.

- One minimizes: $AIC(k) = -2l(\hat{\theta}_k) + 2k$

- AIC has been reported to consistently overestimate even the order of simple linear AR models.
The Model Selection Problem

Maximum Likelihood: \( \log P(x \mid \theta) \),

\[ x = [x_1, \ldots, x_N] \]

\[ \theta = [\theta_1, \ldots, \theta_k] \]

And if \( k \) is not given? \( k \rightarrow N \) failure!

Rissanen: All models can be regarded as codes

The best model is the one corresponding to the shortest encoding of the data
How to encode the data

• Let us suppose we have a random process generating \((Y_t, Z_t)\) in \(\mathbb{R} \times \mathbb{R}^d\), and a relationship:

\[ y_t = F(Z_t) + \varepsilon_t \]

with \(\varepsilon_t\) i.i.d. with finite variance.

• If \(P\) is a probability distribution on the data, a particular realization \(x = \{(y_t, Z_t)\}, \ t = 1, \ldots, N\), can be coded with a minimum code length of

\[- \log_2 P(x)\] bits
How to transmit the data

• One actually transmits a two-part code.

• The first part is: \( \hat{F}(X) = G(X, \theta) \)

• This will allow the receiver to decode the second part, i.e. the encoded data.

• The total code length is

\[
L(x, \theta) = -\log P(x \mid \theta) + L(\theta)
\]
How to encode a model

- The elements of $\theta$ are real numbers, so one as to truncate them in order to transmit them in a finite code length.

$$1011.011 \rightarrow 1011011$$

Accuracy: $\delta = 2^{-3}$

One has to know how to encode integers

- Rissanen’s approach assumes a prior distribution on the parameters.
Universal Prior for Integers

\[ L(x, \theta) = -\log P(x \mid \theta) + L(\theta) \]

Bayes:
\[ P(x \mid \theta) = \frac{P(x, \theta)}{P(\theta)} \]

\[ P(x \mid \theta) = 2^{L(\theta)} \cdot 2^{-L(x, \theta)} \]

Distribution!
Prefix Codes

Example:  \( C(a) = 0 \quad C(c) = 110 \)
\( C(b) = 10 \quad C(d) = 111 \)

1011000111010 \( \rightarrow \) bcaadab
Kraft Inequality

\[ \sum 2^{-L(i)} \leq 1 \]
OPTIMAL CODES

- The average length:

\[ L = - \sum_{i} p_i L(i) \]

of a prefix code is bounded below by:

\[ H = - \sum_{i} p_i \log_2(p_i) \]

\( H \) the entropy of the code.
• Let us define:

\[ n = 9 \quad \implies \quad b(n) = 100 \quad \implies \quad \langle b \rangle = 100001 \]

\[ |b(n)| = \left\lfloor \log_2 2n \right\rfloor \]

• One obtains a prefix code with:

Code word:  \( \langle b(|b(n)|) \rangle b(n) \)

Code length:  \( |b(n)| + 2|b(|b(n)|)| \)
Coding of Integers – first approach

Example:  \( n = 9 \)

\[ b(9) = 1001 \]

\[ b(|b(9)|) = 100 \]

\[ \langle b(|b(9)|) \rangle = 100001 \]

\[ \langle b(|b(9)|) \rangle b(9) = 1000011001 \]
## Coding of Integers – second approach

Kraft Inequality does not hold for pure binary representation

Idea: preamble which is code string length

How to encode the code string length?

Attach preamble which is the length of the code string length ...

| $l_3$ of $l_2$ | $l_2$ of $l_1$ | $l_1$ of string | string |
Decoding rule: Integer $j$ announces the next length in the following $j+1$ positions.

Example: $101001000001110\ldots$
• If we define: \( \log^* n = \log_2 n + \log_2 \log_2 n + \ldots \)

• It can be proven that:

\[
\sum 2^{-\log^*_2(n)} = c
\]

• And with:

\[
L_0(n) = \log^*_2 n + \log_2 c
\]

\[
Q(n) = 2^{-L_0(n)}
\]

is a universal prior for integers
Prior based on computer representation

- Each parameter $\theta_j$ can be expressed with the normalized floating-point binary number:

$$0.1a_1a_2\cdots\times2^m_j$$

- If it is truncated to $\bar{\theta}_j = 0.1a_1a_2\cdots a_{n_j}\times2^m_j$ then the error is at most:

$$\delta_j = 2^{-n_j}$$
Prior based on computer representation

• As a consequence, the total code length for $k$ parameters is:

$$L(\bar{\theta}) = \sum_{j=1}^{k} L_0(1/\delta_j) + \sum_{j=1}^{k} L_0(\log(2 \max\{\bar{\theta}_j, 1/\bar{\theta}_j\}))$$

• The log log... terms vary slowly, and, most of the time, the exponent cost can be fixed at $m$ bits. Then, this expression is well approximated by:

$$\bar{L}(\bar{\theta}) = \sum_{j=1}^{k} \log \frac{\gamma}{\delta_j} \quad \text{with} \quad \gamma = 2^m$$
Minimum Description Length

• The total description length is:

\[ L(x, \bar{\theta}) = L(x|\bar{\theta}) + L(\bar{\theta}) \]

• However, it should not be too far, and:

\[ L(x, \bar{\theta}) \leq L(x, \hat{\theta}) + \frac{1}{2} \delta^T Q \delta \]

with \( Q = D_{\theta \theta} L(x|\theta)|_{\theta = \hat{\theta}} \)
Minimum Description Length

- We obtain:

\[ L(x, \bar{\theta}) \leq L(x|\hat{\theta}) + \frac{1}{2} \delta^T Q \delta + k \log \gamma - \sum_{j=1}^k \log \delta_j \]

- Minimization over \( \delta \) gives:

\[ (Q\delta)_j = 1/\delta_j \quad \text{for each } j \]

- The bound on minimum description length is then:

\[ \text{MDL}(k) = L(x|\hat{\theta}) + \left( \frac{1}{2} + \log \gamma \right) k - \sum_{j=1}^k \log \left( \delta_j \right) \]
MDL for pseudo-linear models

• These models have the general form:

\[ G(z, \theta) = \sum_{i=1}^{m} \theta_i f_i(z) \]

• Typically, one assumes the errors are normally distributed, and ML estimation is simply least-squares estimation.

\[
\min \| y - V\theta \| \\
y = [y_1, \ldots, y_N]^T \quad \theta = [\theta_1, \ldots, \theta_m]^T \quad V_{ij} = f_j(z_i)
\]
MDL for pseudo-linear models

• Under these assumptions:

\[
MDL(k) = \frac{1}{2} N \cdot \ln \hat{\sigma}_e^2 + k \left( \frac{1}{2} + \ln \gamma \right) - \sum_{j=1}^{k} \ln \hat{\delta}_j + C
\]

• For \( N \) large, it is possible to show that:

\[
Q = D_{\theta \theta} L(x|\theta)|_{\theta = \hat{\theta}} \approx \text{diag} \left( \frac{1}{N} \right)
\]

It corresponds to the well-known fact that the parameter estimates are asymptotically Gaussian, non-correlated, with variance \( 1/N \).
• This gives:

\[ \hat{\delta}_j \approx \sqrt{N} \]

• In these conditions also, the term \( k/2 \) becomes small with respect to the other terms, and one ends up with the simplified MDL criterion (a factor 2 is used):

\[ MDL(k) = N \cdot \ln \hat{\sigma}_e^2 + (k + 1) \cdot \ln N \]
Example: MA(5)

\[ y(n) = \sum_{m=1}^{5} a_i x(n - i) + e(n) \]

\[ e(n) \sim N(0, \sigma^2) \]
Example: MA(5)

\[ \sigma_e^2 = 0.01 \]

\[ \sigma_e^2 = 1.0 \]
Orthogonal least squares method:

\[ y(n) = \sum_{m=0}^{M} g_m w_m(n) + e(n) \]

\[ \hat{y}_1(n) = g_1 w_1(n) \]

\[ \hat{y}_2(n) = g_1 w_1(n) + g_2 w_2(n) \]

\[ \hat{y}_M(n) = g_1 w_1(n) + g_2 w_2(n) + \ldots + g_M w_M(n) \]

\[ w_m(n) = x(n-i) \]

\[ w_m(n) = x(n-i) x(n-j) \]

\[ w_m(n) = x(n-i) x(n-j) x(n-k) \]
Model Selection, Polynomials of Order 3

Signal length vs. Number of parameters for high noise and low noise conditions.

- **x**: MDL without accuracy computation
- **o**: MDL with accuracy computation

**High Noise**:
- Increasing signal length shows an increase in the number of parameters needed.

**Low Noise**:
- The number of parameters remains relatively constant across signal lengths.

Signal Processing Laboratory
Swiss Federal Institute of Technology, Lausanne
Model Selection, Polynomials of Order 2

\[ \text{x : MDL without accuracy computation} \]
\[ \text{o : MDL with accuracy computation} \]
\[ \text{* : AIC} \]
Model Selection, Linear Models

![Graph showing model selection results for high and low noise scenarios.]

- **X**: MDL without accuracy computation
- **o**: MDL with accuracy computation
- ***: AIC

Signal Processing Laboratory
Swiss Federal Institute of Technology, Lausanne
Genetic Algorithm (GA)

- Small population
- Binary coding of regressors
- Three operator GA
  - Reproduction of the fittest
  - Mutation
  - Crossover
- Fitness function
  - Minimum description length (MDL)
Genetic Algorithm (GA)

\[ M \]

\[
\begin{bmatrix}
1 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 1 \\
1 & 1 & 0 \\
1 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 1 \\
1 & 1 & 0 \\
1 & 0 & 0 \\
1 & 0 & 0
\end{bmatrix}
\]

1: select regressor
0: disregard regressor
Genetic Algorithm (GA)

- Mutation

10011100 \rightarrow 10011000

- Crossover

\[
\begin{array}{c|c}
10011100 & 10001010 \\
01101010 & 01101010 \\
\end{array}
\]

\[\rightarrow \quad 10001010\]
Genetic Algorithm (GA)

1. Reproduction of the fittest

2. \((N-1)/2\) mutations of the fittest

3. \((N-1)/2\) crossovers on the remaining strings
## Average Number of Generations

\[ y(n) = a_1 y(n-4) + a_2 y(n-8) + e(n) \]

<table>
<thead>
<tr>
<th>Population</th>
<th>Chromosome length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>7.67 ± 3.7</td>
</tr>
<tr>
<td>11</td>
<td>5.84 ± 2.3</td>
</tr>
<tr>
<td>13</td>
<td>6.85 ± 3.3</td>
</tr>
</tbody>
</table>
Average Number of Generations and Evaluations

\[ y(n) = a_1 x(n-3) + a_2 x(n-1)x(n-3) + a_3 x(n-2)x(n-4) + a_4 x(n-3)x(n-5) + e(n) \]

<table>
<thead>
<tr>
<th>Population</th>
<th>Generations</th>
<th>Evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>24.97 ± 8.8</td>
<td>150.82</td>
</tr>
<tr>
<td>9</td>
<td>17.88 ± 6.2</td>
<td>144.04</td>
</tr>
<tr>
<td>11</td>
<td>15.33 ± 5.1</td>
<td>154.30</td>
</tr>
<tr>
<td>13</td>
<td>14.57 ± 4.6</td>
<td>175.84</td>
</tr>
</tbody>
</table>
