Workshop the Minimum Description Length Principle

Exercises with Answers

Paul van Leeuwen (paul2.vanleeuwen@devolksbank.nl)
Eelko Ubels (eelko.ubels@devolksbank.nl)

13 June 2023

Introduction

Exercises are meant as an introduction to the Minimum Description Length (MDL) principle and are by no means meant to stop here. They are merely for illustrative purposes; in (academic) applications they are more advanced. For a decent introduction into MDL, see [1]. Presumed knowledge: basic calculus, presentation of the MDL principle. Results of exercises could be used in subsequent exercises. Hints are given on separate pages after the exercises.

1 Morse Code

In 1840 Samuel Morse faced a similar problem as we do: how to code the Western alphabet $\mathbb{B} = \{\mathrm{A}, \mathrm{B}, \ldots, \mathrm{Z}, 0, 1, \ldots, 9\}$ such that on expectation the smallest amount of bits is needed for whatever message. Every symbol is a concatenation of *dits* (one hit with the sending device with a fixed duration represented by a dot $\cdot$) and *dahs* (three times the duration of a *dit* represented by a bar $-$). For example, the letter A is a dit followed by a dah ($\cdot -$). See Figure 1 for the complete mapping of the $\mathbb{B}$ to the Morse symbols.[1]

1. You can assign a single dit to just one symbol. What letter would you recommend?

   Answer The letter E for that letter is observed most frequently.

2. A simple alternative is to have a fixed code length for every symbol. How many bits would we need to represent all elements of $\mathbb{B}$ in a binary way?

   Answer In total there are 36 possible symbols we may encounter. That requires $\lceil \log_2(36) \rceil = 6$ positions for all elements of $\mathbb{B}$. For example, we could code A by 000000, B by 100000, etc. Note that not all possible sequences of ones and zeros of length 6 are assigned to elements of $\mathbb{B}$. For that $\mathbb{B}$ would need to have $2^6 = 64$ different symbols.

3. Samuel did not choose for this simple alternative but aimed to minimise the *expected* code length. Why?

   Answer We do not now what message we could expect upfront. If that were the case we could do even better than the Morse code. For example, if we know that only two messages `Hello, world!` and `Bye, world!` can be communicated, we need only 1 bit. But Morse code is designed to accommodate all possible concatenations of symbols from $\mathbb{B}$.

[1] Retrieved from [2].

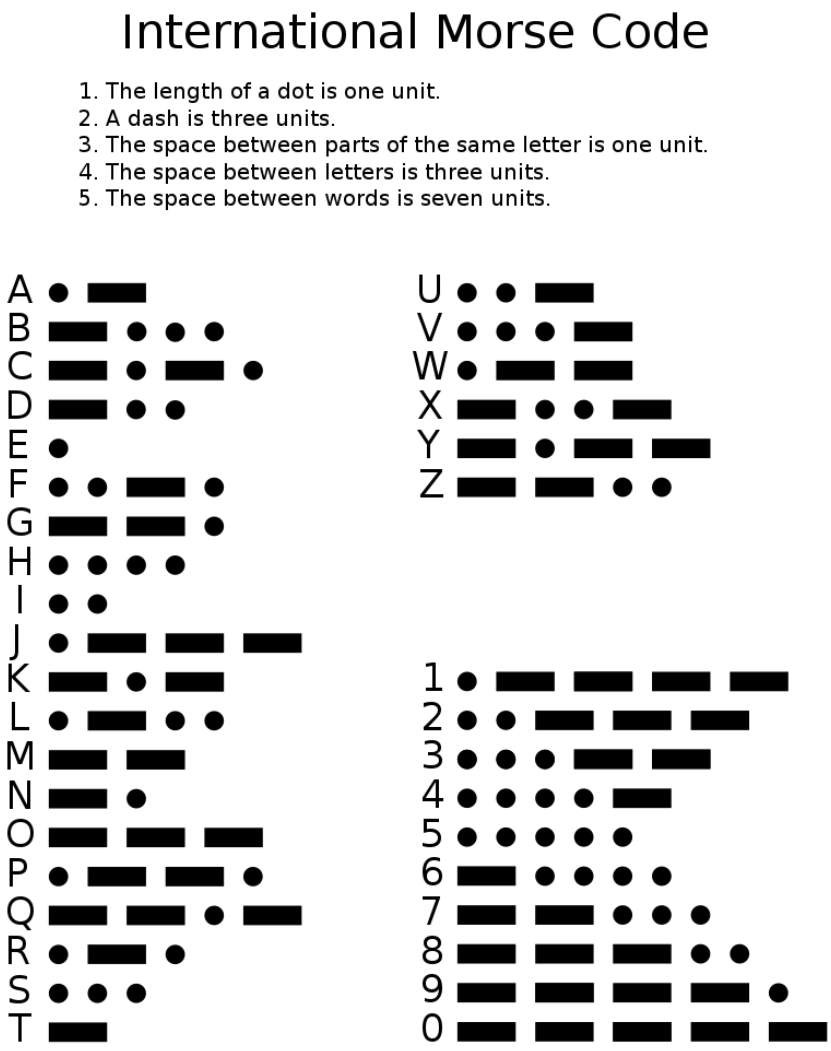


Figure 1: International Morse Code of the Western alphabet and the 10 digits.

2 Design Your Own Code

Suppose you are given the alphabet $\mathbb{B} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$ with corresponding observed frequencies $(15, 7, 6, 6, 5)$.[2] So in total 39 symbols are communicated. In this exercise you will design a coding scheme $C\colon \mathbb{B} \mapsto \{0,1\}^m$ with at most m symbols and $m \in \mathbb{N}_+$ chosen by you. A possible mapping of the symbol B could be $C(\mathrm{B}) = 101$. Note that in the exercises below you do not need to stick with $C(\mathrm{B}) = 101$.

1. Code the alphabet $\mathbb{B}$ using a uniform coding scheme: every symbol uses the same number of bits.

   Answer In total there are 5 possible symbols we may encounter. That requires $\lceil \log_2(5) \rceil = 3$ positions for all elements of $\mathbb{B}$. For example, we could code A by 000, B by 100, etc. Note that not all possible sequences of ones and zeros of length 3 are assigned to elements of $\mathbb{B}$. An alphabet that would utilise all $2^3 = 8$ possible sequences has 8 different symbols.

2. What is the expected code length using this coding scheme?

   Answer Also 3, because whatever the probability is a certain symbol of $\mathbb{B}$ will show up, the code length is always 3.

Whatever concatenation of zeros and ones (or dits and dahs in Morse code) you assign to every symbol, the resulting code should be *prefix*: reading from left to right always results in the same interpretation. In practice this means that, reading from left to right, no code should be a part of another code. Let $C\colon \mathbb{B} \mapsto \{0,1\}^m$ be the mapping from an element of the alphabet $\mathbb{B}$ to a concatenation of at most m zeros and ones. An example of bad coding is the following: $C(\mathrm{A}) = 10$ and $C(\mathrm{C}) = 100$ is not prefix since the 10 may symbolise A or C followed by a 0.

3. Is Morse-code a prefix code? See Figure 1 for the complete mapping of the $\mathbb{B}$ of Exercise 1 to the Morse symbols.

   Answer No. For example, the letter E (Morse-code: $\cdot$) is the first symbol of the letter A (Morse-code: $\cdot -$). Therefore we cannot know whether the dot refers to the letter E or the beginning of the letter A.

4. We can do better when the probability a symbol shows up is different from another. Assume the observed frequencies are representative for all future messages to be coded. What would be a more efficient coding scheme? Make sure the code is prefix.

   Answer (Hint) A possible optimal code is found by making use of the Huffman code: $C(A) = 0$, $C(B) = 100$, $C(C) = 101$, $C(D) = 110$, $C(E) = 111$. The expected code length is then given by

   $$\frac{15}{39} \cdot 1 + \frac{7}{39} \cdot 3 + \frac{6}{39} \cdot 3 + \frac{6}{39} \cdot 3 + \frac{5}{39} \cdot 3 = 2\frac{9}{39}$$

   bits, an improvement of $\frac{30}{39}$ bits over the uniform coding scheme.

3 From Code Length to Probabilities

Suppose we have a message of length $k \in \mathbb{N}_+$ symbols from the binary alphabet $\{0,1\}$. Let p_i, for $i = 1, \ldots, k$ be the probability that symbol i will occur with $\sum_{i=1}^{k} p_i = 1$ and let $\ell_i \in \mathbb{R}_+$ be the corresponding code length; the total code length is L. Then the expected code length is

$$\mathbb{E}[L] = \sum_{i=1}^{k} p_i \ell_i$$

[2] Example taken from [3].

The expected code length, using the Shannon-Fano code $\ell_i = -\log_2 p_i$, is given by

$$E = \sum_{i=1}^{k} -p_i \log_2 p_i \tag{1}$$

E is also known as the entropy. We could code the entire message symbol by symbol, processing them from left to right, without taking the complete sequence of symbols into account. This is as if the complete message of length k can be broken down into k separate submessages. Alternatively, we could assign a symbol to every possible sequence of $\{0,1\}^k$.

1. Suppose that we code the total message symbol by symbol. Explain why the choice of $\ell_i = -\log_2 p_i$ is not always realistic to accurately represent the code length of all corresponding p_i.

   Answer $-\log_2 p_i$ does not necessarily result in an integer while the length of a code is always an integer; so $-\log_2 p_i$ should be rounded to the next nearest integer. For example, $p_1 = \frac{1}{4}$ results in code length $-\log_2 p_1 = 2$ and $p_1 = \frac{1}{\pi}$ results in code length $-\log_2 p_1 = 1.65$, and also requires 2 bits.

2. (Hint) Choose $\ell_i = \lceil -\log_2 p_i \rceil$. Derive an upper bound for $\mathbb{E}[L]$ in terms of E.

   Answer

$$\mathbb{E}[L] = \sum_{i=1}^{k} p_i \ell_i = \sum_{i=1}^{k} p_i \lceil -\log_2 p_i \rceil \leq \sum_{i=1}^{k} p_i(-\log_2 p_i + 1) = \left(\sum_{i=1}^{k} -p_i \log_2 p_i \right) + \sum_{i=1}^{k} p_i = E + 1.$$

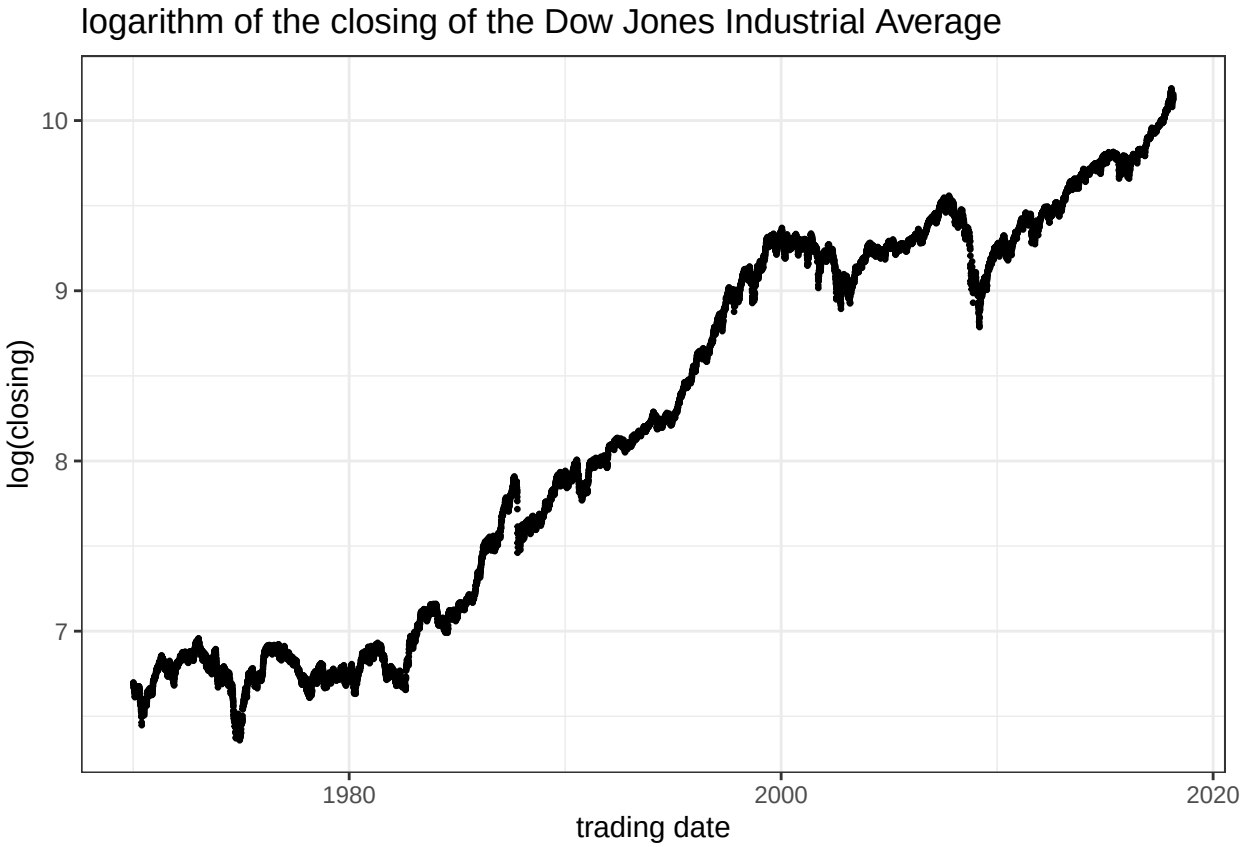
3. Using $\ell_i = -\log_2 p_i$ we have a connection between the probability that event i occurs, where event i is represented by symbol i, and code length ℓ_i. Although this choice of ℓ_i does not give realistic code lengths, explain why it is useful after all.

   Answer MDL serves the purpose of, among else, model selection. For that we do not need a more efficient coding scheme than the Shannon-Fano coding scheme that assigns code length $\lceil -\log_2 p_i \rceil$. We just need the connection that the code length of an event i with probability p_i of occurring equals $-\log_2 p_i$. Since either $\lceil -\log_2 p_i \rceil$ or $-\log_2 p_i$ results in, apart from a constant, the same expected code length, both are fine to use in the search of the optimal model.

4 From Data to Parameter Estimation

In this exercise we will use the same approach as in [4]. They model the Dow Jones Industrial Average (DJIA) using different models where they choose the winning model using MDL. For every trading day the closing of the DJIA is denoted by $P_t \in \mathbb{R}_+$ where t runs from $t_0 =$ January 1970 until $T =$ 12 June 1995, i.e. 6,430 trading days.[3] See the next figure for a visualisation of P_t.

[3] Note that [4] start at June 1962 but that data is unavailable via standard retrieval methods. Therefore the end date 12 June 1995 is chosen such that the total same number of trading days over this period is the same as in [4].



Take, for $t \geq 1$, the log daily return $R_t = P_t - P_{t-1}$ and the volatility $V_t = 0.9V_{t-1} + 0.1R_t^2$ with V_0 the variance of the series P_t. R_t has a corresponding indicator: 1 (0) when $P_t > P_{t-1}$ ($P_t < P_{t-1}$), analogously for V_t. This results in two binary strings of length 6,430 - 1 = 6,429. R_t has 3,166 (49.25%) ups and V_t has 1,964 (30.55%) ups. As with Morse code, the sender and receiver need to agree upfront how the events are coded to a concatenation of zeros and ones. We do not have prior knowledge on the outcomes of the indicators of R_t and V_t.

1. As a baseline we consider the uniform coding scheme. How many bits do we require to code the ups and downs of R_t? And how many for the ups and downs of V_t?

   Answer The length of the binary string, so 6,429 bits for both. For every day we reserve a 1 or 0 and, depending on the outcome, we record a 1 in case of an up and a 0 in case of a down.

2. As before, we can do better when the probability of one event is sufficiently different from $\frac{1}{2}$. For what time series R_t or V_t do you expect, *a priori*, to realise the highest compression?

   Answer The time series with the probability on an up that deviates most from $\frac{1}{2}$. The probability of an up for R_t is 0.4925 and for V_t it is 0.3055 so we expect the highest improvement for V_t.

Two-part MDL is, among the possible MDL methods, a simplistic method to measure the level of compression. The two refers to two stages: (i) describe the data description mechanism H with code length $L(H)$, and (ii) given H describe the data D with code length $L(D|H)$. The sender, or coder, agrees with the receiver, or decoder, to model the ups and downs of R_t and V_t with a Bernoulli model. So before the data can be coded, first the data description mechanism needs to be communicated from the sender to the receiver, followed by the data, i.e. the concatenation of zeros and ones.

3. Derive the maximum likelihood estimator $\hat{p}$ for p.

Answer Define by X_i, $i = 1, \ldots, n$ for $n \in \mathbb{N}$ observations, the random variable with realisation $x_i \in \{0, 1\}$ and with probability $p = \mathbb{P}[X_i = 1]$ on success. The likelihood that we observe k successes out of n observations is given by

$$L = \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} = p^{\sum_{i=1}^{n} x_i}(1-p)^{\sum_{i=1}^{n} 1 - x_i} = p^k(1-p)^{n-k}$$

Take the logarithm on both sides to be able to derive an analytical solution, i.e.

$$\log L = k \log(p) + (n-k)\log(1-p) \tag{2}$$

and take the derivative with respect to p,

$$\frac{\partial}{\partial p} \log L = \frac{\partial}{\partial p}\big[k \log(p) + (n-k)\log(1-p)\big] = \frac{k}{p} - \frac{n-k}{1-p}$$

Equating $\frac{\partial}{\partial p} \log L$ to 0 results in $\hat{p} = k/n$. To confirm that for that choice of p we indeed have the maximum likelihood estimator, the second derivative of $\log L$ needs to be negative for $\hat{p}$. Indeed,

$$\frac{\partial^2}{\partial p^2} \log L = -\frac{k}{p^2} + \frac{n-k}{(1-p)^2}$$

is strictly negative for all k and n when $p = k/n$.

4. (Hint) How many bits do you need to communicate H, the maximum likelihood estimate $\hat{p}$, using a uniform coding scheme?

   Answer Since $\hat{p} = k/n$ and k can take $n + 1$ different values, we need to code an alphabet $\mathbb{B}$ of cardinality $n+1$. With the uniform coding scheme we need $\lceil \log_2(n+1) \rceil$ positions to code all elements of $\mathbb{B}$. So both for R_t and V_t we need $\lceil \log_2(6430) \rceil = 13$ bits. Note that the uniform coding scheme is chosen to not *a priori* express preference for a certain choice for k. Because that would imply we assign a non-uniform prior probability distribution to k hence, we have prior knowledge that justifies that choice. However, that would contradict with what is given above.

5. (Hint) Given that $\hat{p}$ is communicated, how many bits do you need to communicate the ups and downs of R_t and V_t, i.e. $L(D|H)$?

   Answer $L(D|H)$ is the minus of the log-likelihood of Equation (2). We have k times an up with code length $-\log_2\left(\frac{k}{n}\right)$ and $n-k$ times a down with code length $-\log_2\left(1-\frac{k}{n}\right)$, together resulting in

$$L(D|H) = \left\lceil -k \log_2\left(\frac{k}{n}\right) - (n-k)\log_2\left(1-\frac{k}{n}\right) \right\rceil$$

   For R_t we have $k = 3166$ and $\hat{p} = \frac{k}{n} = \frac{3166}{6429} = 0.4925$, so we need

$$\big\lceil -3166 \log_2(0.4925) - 3264 \log_2(0.5077) \big\rceil = 6428$$

   bits. For V_t we have $k = 1964$ and $\hat{p} = \frac{k}{n} = \frac{1964}{6429} = 0.3055$, so we need

$$\big\lceil -1964 \log_2(0.3055) - 4406 \log_2(0.6945) \big\rceil = 5709$$

   bits.

6. Does that result in an improvement relative to the number of bits required for the uniform coding scheme as derived in (1)?

   Answer The total required code length is given by

$$L(H, D) = L(H) + L(D|H) = \lceil \log_2(n) \rceil + \left\lceil -k \log_2\left(\frac{k}{n}\right) - (n-k)\log_2\left(1-\frac{k}{n}\right) \right\rceil$$

   So for R_t we need $13 + 6428 = 6441$ bits and for R_t we need $13 + 5709 = 5722$ bits. In conclusion, in comparison to the uniform coding scheme that demanded 6429 bits, we have gained nothing for R_t, even lost some bits, but the improvement for V_t is substantial.

5 Markov Modelling

An improvement over the Bernoulli model could be a Markov model; more precise: a time-discrete Markov chain. In this way we are able to capture, if present, time-dependency in the ups and downs of R_t and V_t. Let $C_t \in \{0,1\}$ denote the event that $R_t > R_{t-1}$ with realisations $c_t \in \{0,1\}$. The Markov property states that the probability of an up on the next day only depends on the realisation on the previous k values: $\mathbb{P}[C_{t+1} = 1 | C_t = c_t, \ldots, C_{t-k} = c_{t-k}]$ for $t = 1, \ldots, T$ and $k \geq 1$; when the probability that $C_{t+1} = 1$ occurs is independent of previous values of C_t, we have $\mathbb{P}[C_{t+1} = 1$ for $k = 0$. The probability transition matrix $\mathbf{P}$ for k lags is a matrix of size $\left(2^k \times 2\right)$ where $\mathbf{P}_{ij}$ is the probability at row i and column j with $i = 1, \ldots, 2^k$ and $j = 1, 2$. In this case, i ranges over all possible routes the lagged target values can take and j indicates whether the target value on the next day is a 0 ($j = 1$) or a 1 ($j = 2$). By definition, the probabilities of one row over all possible columns should be 1:

$$\sum_{j=1}^{2} \mathbf{P}_{ij} = 1$$

For example, for $k = 2$ we have

$$\mathbf{P} = \begin{pmatrix} p_{00\to 0} & p_{00\to 1} \\ p_{01\to 0} & p_{01\to 1} \\ p_{10\to 0} & p_{10\to 1} \\ p_{11\to 0} & p_{11\to 1} \end{pmatrix}$$

1. Argue that the Bernoulli model is a nested model of the k^{th}-order Markov model family for $k \in \mathbb{N}$. In other words, what is k for the Bernoulli model?

   Answer A Bernoulli model is a memoryless Markov model, i.e. $k = 0$. In general, the probability transition matrix $\mathbf{P}$ of a k^{th}-order Markov model has 2^k rows and k columns.

2. In what case would a higher-order Markov model be superior over a Bernoulli model?

   Answer When there is autocorrelation present. For then the memory comes into play; with the Bernoulli model that is not possible for the probability on a success cannot be dependent on previous values.

3. Study the autocorrelation function of $R_t - R_{t-1}$ and $V_t - V_{t-1}$. In Matlab you may want to use the function `autocorr`. The realisations of R_t and V_t can be extracted from the Excel-file that was sent to you by mail by making use of the Matlab-function `readtable`.

   Answer After retrieving the data from Excel and executing the Matlab-function `autocorr` on $R_t - R_{t-1}$ and $V_t - V_{t-1}$ we get, with a lag of 1, autocorrelation values of -0.2687 and -0.0323, respectively. The high (absolute) autocorrelation value for R_t is a strong indicator that the first-order Markov model yields a better data compression than the lower (absolute) autocorrelation value for V_t.

4. (Hint) Derive the maximum likelihood vector of model parameters for the first-order Markov model.

   Answer The probability transition matrix $\mathbf{P}$ for $k = 1$ lags is given by

   $$\mathbf{P} = \begin{pmatrix} p_{0\to 0} & p_{0\to 1} \\ p_{1\to 0} & p_{1\to 1} \end{pmatrix}$$

   As before, define by X_i, $i = 1, \ldots, n$ for $n \in \mathbb{N}$ observations, the random variable with realisation $x_i \in \{0,1\}$. The probability on success is thus dependent on the realisation of the previous value:

   $$p_{0\to 0} = \mathbb{P}[X_i = 1 | X_{i-1} = 0] \quad \text{and} \quad p_{1\to 0} = \mathbb{P}[X_i = 1 | X_{i-1} = 1]$$

   Let $k_{\ell\to 1}$ ($k_{\ell\to 0}$) be the number of successes (failures) conditional on the previous value $\ell = 0, 1$. Then the likelihood is given by

   $$L = \frac{1}{2} \prod_{i=2}^{n} \mathbb{P}[X_i | X_{i-1}] = \frac{1}{2} p_{0\to 0}^{k_{0\to 0}} p_{0\to 1}^{k_{0\to 1}} p_{1\to 0}^{k_{1\to 0}} p_{1\to 1}^{k_{1\to 1}} \tag{3}$$

where $\frac{1}{2}$ in front of the product is the (unconditional) probability on a success for $i = 1$. The probabilities in each row sum up to one, resulting in

$$p_{0\to1} = 1 - p_{0\to1} \quad \text{and} \quad p_{1\to1} = 1 - p_{1\to0}$$

Substitution in Equation (3) results in

$$L = \frac{1}{2} p_{0\to0}^{k_{0\to0}} (1 - p_{0\to0})^{k_{0\to1}} p_{1\to0}^{k_{1\to0}} (1 - p_{1\to0})^{k_{1\to1}}$$

Taking the logarithm yields

$$\log_2 L = -\log_2(2) + k_{0\to0} \log_2(p_{0\to0}) + k_{0\to1} \log_2(1-p_{0\to0}) + k_{1\to1} \log_2(p_{1\to0}) + k_{1\to1} \log_2(1-p_{1\to0}) \quad (4)$$

and differentiating with respect to $p_{0\to0}$ and $p_{1\to0}$ leads to

$$\frac{\partial}{\partial p_{0\to0}} \log L = \frac{k_{0\to0}}{p_{0\to0}} - \frac{k_{0\to1}}{1 - p_{0\to0}} \quad \text{and} \quad \frac{\partial}{\partial p_{1\to0}} \log L = \frac{k_{1\to0}}{p_{1\to0}} - \frac{k_{1\to1}}{1 - p_{1\to0}}$$

Equating to 0 and isolating $p_{0\to0}$ and $p_{1\to0}$ gives

$$\widehat{p_{0\to0}} = \frac{k_{0\to0}}{k_{0\to0} + k_{0\to1}} \quad \text{and} \quad \widehat{p_{1\to0}} = \frac{k_{1\to0}}{k_{1\to0} + k_{1\to1}}$$

The check that the second derivative of L with respect to $p_{0\to0}$ and $p_{1\to0}$ is negative for all $k_{0\to0}$, $k_{0\to1}$, $k_{1\to0}$, and $k_{1\to1}$ is left for the reader.

5. To code a first-order Markov model, we need to first code the data description mechanism H with code length $L(H)$. Then, given H, give the code length of the data D, which is $L(D|H)$.

   Answer Compared to the Bernoulli model the data description mechanism H consists of two instead of one model parameters: $p_{0\to0}$ and $p_{1\to0}$. The number of model parameters doubles and therefore $L(D|H)$ doubles as well: $L(D|H) = 2\log_2(n+1)$, ignoring rounding errors. So both for R_t and V_t we need $2\lceil\log_2(6430)\rceil = 26$ bits to communicate $\widehat{p_{0\to0}}$ and $\widehat{p_{1\to0}}$.

   Given that $\widehat{p_{0\to0}}$ and $\widehat{p_{1\to0}}$ are known to the receiver, the code length of the data given H, $L(D|H)$, is given by the minus of the log-likelihood $\log_2 L$ in Equation (4). For R_t that becomes

   $$L(D|H) = \lceil -\log_2(2) + 1221\log_2(0.3753) + 2032\log_2(0.6247) + 2032\log_2(0.6439) + 1124\log_2(0.3561)\rceil$$

   or 6070 bits. For V_t that becomes

   $$L(D|H) = \lceil -\log_2(2) + 3039\log_2(0.6837) + 1406\log_2(0.3163) + 1406\log_2(0.7159) + 558\log_2(0.2811)\rceil$$

   or 5701 bits. Compared to the Bernoulli model the first-order Markov model results in a significant data compression of $6428 - 6070 = 358$ (-5.9%) for R_t. For V_t the compression is insignificant. This finding confirms our intuition of Exercise (5.3): the relatively strong autocorrelation of R_t implies that the first-order Markov model is a better predictor than the Bernoulli model, compared to V_t.

6. This two-stage approach is rather simplistic. Give suggestions on how to improve this approach.

   Answer

   - When you have relevant knowledge a Bayesian prior probability distribution could be of help.
   - The higher the arithmetic precision, the more digits are required for each probability, the more bits are required to code each probability.

6 Hints

(2.4)
Hint 1: First calculate the probability a certain symbol occurs by relating the frequency of occurrence to the total number of symbols.

Hint 2: The higher the probability a symbol occurs, the smaller the code length can be. So assign the smallest code, which is thus a concatenation of zeros and ones, to A and the largest code to E.

(3.2) $\lceil -\log_2 p_i \rceil \leq -\log_2 p_i + 1$

(4.4) What possible values could $\hat{p}$ take?

(4.5) We are interested in the expected value of L. $\mathbb{E}[L]$ can be approximated by E, see Equation (1), with $\hat{p} = k/n$ and summing over all n events that end in an up or a down.

(5.4) The Markov model has two model parameters: $p_{0\to0}$ and $p_{1\to0}$ since $p_{0\to1} = 1 - p_{0\to0}$ and $p_{1\to1} = 1 - p_{1\to0}$.

Bibliography

[1] P. D. Grünwald, *The minimum description length principle*. MIT press, 2007.

[2] Wikipedia, "International morse code," 2023-06-13. https://en.wikipedia.org/wiki/Morse_code#/media/File:International_Morse_Code.svg (accessed Jun. 09, 2023).

[3] Wikipedia, "Shannon-fano coding," 2023-06-09. https://en.wikipedia.org/wiki/Shannon%E2%80%93Fano_coding (accessed Jun. 09, 2023).

[4] M. H. Hansen and B. Yu, "Model selection and the principle of minimum description length," *Journal of the American Statistical Association*, vol. 96, no. 454, pp. 746–774, 2001.