Workshop the Minimum Description Length Principle

Exercises

Paul van Leeuwen (paul2.vanleeuwen@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 (Grünwald 2007). 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 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 ($\cdot -$).

1. You can assign a single dit to just one symbol. What letter would you recommend?
2. A simple alternative is to have a fixed code length for every symbol. How many positions would we need?
3. Samuel did not choose for this simple alternative but aimed to minimise the *expected* code length. Why?

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 $(15, 7, 6, 6, 5)$. 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 $C(\mathrm{B}) = 101$.

1. Code the alphabet $\mathbb{B}$ using a uniform coding scheme: every symbol uses the same number of bits.
2. What is the expected code length using this coding scheme?

   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: $C(\text{A}) = 10$ and $C(\text{C}) = 100$ is not prefix since the 10 may symbolise C or B followed by a one.

3. 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.

   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 From Code Length to Probabilities

Let p_i, for $i = 1, \ldots, k$ and $k \in \mathbb{N}_+$ symbols, be the probability that symbol i will occur and let $\ell_i \in \mathbb{N}_+$ be the corresponding code length. Then the expected code length is

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

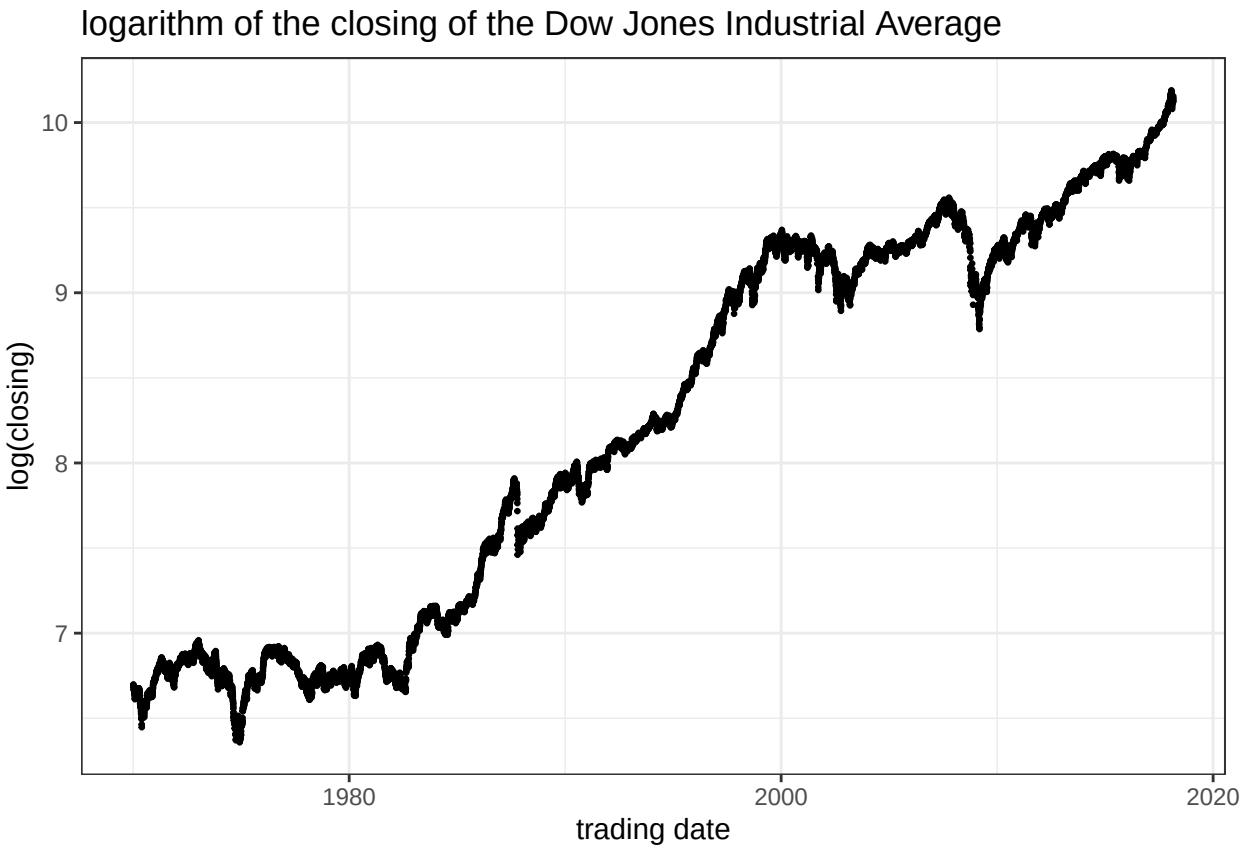
The Shannon-Fano code is given by, using $\ell_i = -\log_2(p_i)$,

$$H = \sum_{i=1}^{k} -p_i \log_2(p_i) \tag{1}$$

1. Explain why this choice for ℓ_i is not realistic to accurately represent the code length of all p_i.
2. Choose $\ell_i = \lceil -\log_2(p_i) \rceil$. Derive an upper bound for $\mathbb{E}[L]$ in terms of H. Hint: $\lceil -\log_2(p_i) \rceil \leq -\log_2(p_i) + 1$.
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.
4. We can achieve expected code length H using efficient coding schemes like Huffman coding and arithmetic coding. Motivate why we are not interested in even more efficient coding schemes when we have $\mathbb{E}[L] \leq H + 1$.

4 From Data to Parameter Estimation

In this exercise we will use the same approach as in (Hansen and Yu 2001). 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 = July 1962 until T = June 1988, i.e. 6,430 trading days. See the next figure for a visualisation of P_t.



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 3,181 (49.49%) ups and V_t has 2,023 (31.47%) 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.

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?

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?

   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, and (ii) given (i) describe the data. 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. Hint: What possible values could $\hat{p}$ take?

4. How many bits do you need to communicate the maximum likelihood estimate $\hat{p}$?

5. Given that $\hat{p}$ is communicated, how many bits do you need to communicate the ups and downs of R_t and V_t? Hint: We are interested in the expected value of L. $\mathbb{E}[L]$ can be approximated by H, see Equation (1), with $\hat{p} = k/n$ and summing over all n events that end in an up or a down.

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

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 of today: $\mathbb{P}[C_{t+1} = 1 | C_t = c_t]$.

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?
2. In what case would a higher-order Markov model be superior over a Bernoulli model?
3. Study the autocorrelation function of R_t and V_t. 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.
4. To code a second-order Markov model, we need to first code the

6 Hints

Bibliography

Grünwald, Peter D. 2007. *The Minimum Description Length Principle*. MIT press.

Hansen, Mark H, and Bin Yu. 2001. "Model Selection and the Principle of Minimum Description Length." *Journal of the American Statistical Association* 96 (454): 746–74.