The Momentum Problem in MDL and Bayesian Prediction

Tim van Erven

May 4, 2006

Master's Thesis

written by

Tim Adriaan Lambertus van Erven
(born April 20, 1982 in Eindhoven, the Netherlands)

under the supervision of **dr. P. D. Grünwald** and the co-supervision of **drs. S. de Rooij**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Artificial Intelligence

at the *Universiteit van Amsterdam*.

Date of the public defense:
May 29, 2006

Members of the Thesis Committee:
Dr. P. van Emde Boas
Dr. P. H. Rodenburg
Dr. P. D. Grünwald
Drs. S. de Rooij

FACULTEIT DER NATUURWETENSCHAPPEN, WISKUNDE EN INFORMATICA

Preface

"Prediction is very difficult, especially about the future."
(attributed to Niels Bohr)

Introduction and Outline

The Minimum Description Length (MDL) principle provides a powerful philosophy for learning from observations of the past [Grünwald et al., 2005; Rissanen, 1989]. It equates learning with compressing the observational data. As is common in science, there may be multiple contending explanations, or *models*, for the data. In this thesis we investigate an application of the MDL principle to prediction of the future when there are at least two such models. We will show that the regular, commonly used form of MDL can behave suboptimally and present a refinement of regular MDL that we call the *Switch-Point procedure*. Being based on data compression, the Switch-Point procedure may still be considered an application of the MDL principle, although it differs from the way in which MDL is usually applied. For the convenience of readers with a background in Bayesian statistics, we give an interpretation of the regular MDL procedure as an instance of Bayesian Model Averaging (BMA). As a consequence our results on MDL transfer to BMA directly.

Our first contribution is to identify the *momentum phenomenon*, which arises when one model enables the most accurate predictions of the future given few observations of the past, but predictions based on another model become more accurate when more data are collected. Essentially, this may happen whenever the models themselves represent compound explanations.

The momentum phenomenon will *not* occur, for example, if one model, $\mathcal{M}_0$, represents the conjecture that the data come from repeated tosses of a biased coin with probability 3/5 of coming up heads, and the other model, $\mathcal{M}_1$, describes the data as tosses of a coin with probability 4/7 of coming up heads. It *can* occur, however, if $\mathcal{M}_1$ were to represent the hypothesis that the data come from a coin with *unknown* probability p of coming up heads. This latter model basically combines all the specific explanations "the probability of coming up heads is 4/7" into the compound explanation "the probability of coming up heads may be any fixed value p". The momentum phenomenon can occur, in that case, if the relative frequency of heads in the data converges to some number f, which is close to, but not equal to 3/5. If this happens, then for few observations of the past the slightly incorrect, but specific model $\mathcal{M}_0$ will enable the best predictions, but when more data are collected predictions based on the correct, but vague model $\mathcal{M}_1$ become more accurate. The reason is that the predictor based on $\mathcal{M}_1$ has to learn the value of the unknown parameter p. When little data is available, its estimate of p will necessarily be poor and may sometimes be further away from f than 3/5. However, the more data becomes available, the better p can be estimated. Therefore, at some point the estimate of p will get closer to f than 3/5 and, from that point on, predictions based on $\mathcal{M}_1$ will be more accurate than those based on $\mathcal{M}_0$.

Though the momentum phenomenon has been previously recognised in a somewhat implicit manner, its consequences have not before been well understood. In particular, it is well known that, in cases such as the one described above, regular MDL predictions will tend to resemble predictions based on model $\mathcal{M}_0$ when few observations are available, and start to resemble predictions based on model $\mathcal{M}_1$, with the unknown parameter, when sufficient data are collected. However, it turns out that the number of observations at which MDL starts predicting in accordance with $\mathcal{M}_1$ can be much larger than the number of observations at which predictions based on $\mathcal{M}_1$ become more accurate than those based on $\mathcal{M}_0$. Thus, there is a certain inertia in the behaviour of regular MDL. We will argue that this is a consequence of the fact that the design of regular MDL procedures does not take the momentum phenomenon into account.

As our second contribution, we develop the Switch-Point procedure, which is a refinement of regular MDL that is designed with the momentum phenomenon in mind. The idea of the Switch-Point procedure is as follows. Let $\mathcal{M}_0$ and $\mathcal{M}_1$ be the same models as above. With each model $\mathcal{M}_j$ we associate a predictor P_j that, given each initial data sequence $x_1, \cdots, x_n$,

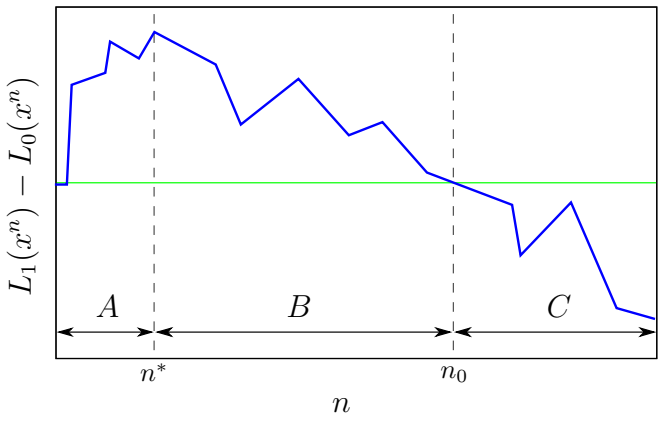


Figure 1: A sketch of the difference in accumulated prediction error on n observations between the predictor for model $\mathcal{M}_1$ and the predictor for model $\mathcal{M}_0$ as a function of n.

makes a prediction of the value that will be observed for the (as yet unseen) datum x_{n+1} according to $\mathcal{M}_j$. Suppose that, when x_{n+1} is actually observed, we have some way of measuring the error of the prediction of P_j for x_{n+1}. Then, *before* seeing x_{n+1}, we can assess the quality of model $\mathcal{M}_j$ on data x_1, $\cdots$, x_n by examining the *accumulated prediction error*, $L_j(x^n)$, of predictor P_j. This is the prediction error made by P_j on x_1, plus the prediction error made on x_2 by P_j based on x_1, plus the prediction error on x_3 based on x_1 and x_2, and so on up to the prediction error made by P_j on x_n based on x_1, $\cdots$, x_{n-1}. The prediction for x_{n+1} made by regular MDL based on observations x_1, $\cdots$, x_n always resembles that of the model $\mathcal{M}_j$ that has achieved the smallest accumulated prediction error on these n observations. Figure 1 shows a rough sketch of the difference in accumulated prediction error between the predictors for model $\mathcal{M}_1$ and model $\mathcal{M}_0$ as a function of the number of observations, n, for a typical case of the momentum phenomenon. We can see that predictions based on $\mathcal{M}_0$ are most accurate in region A since the difference in accumulated prediction error between the models is increasing. In region B the predictor based on model $\mathcal{M}_1$ is already better than the predictor based on $\mathcal{M}_0$. However, it is not until after n_0 observations that MDL *switches* its preference to $\mathcal{M}_1$. This suggests that regular MDL can be improved on by switching to $\mathcal{M}_1$ earlier, namely when the difference in accumulated prediction error is at its maximum. The Switch-Point procedure is an attempt to identify the number of observations, n^*, at which this maximum is achieved and to construct a predictor whose predictions start resembling those of $\mathcal{M}_1$ at a number of observations near n^*, rather than n_0. It will be shown that the Switch-Point procedure can never predict much

worse than regular MDL, but may predict significantly better when the momentum phenomenon occurs. We conclude that the momentum phenomenon can be exploited to improve predictive accuracy compared to regular MDL. This leads to our most important insight, namely that for regular MDL the momentum phenomenon should be considered a momentum *problem*.

Prediction with multiple models is closely related to *model selection*, which is the task of selecting a single model to explain the data. In model selection regular MDL always selects the model with smallest accumulated prediction error. Preferably, the selected model should be the best predictor of future data, which makes model selection closely related to prediction with multiple models. But consider again region B in Figure 1. In this region it is unclear which model should be selected. Has the data been explained best by model $\mathcal{M}_0$, which has enabled us to predict the data most accurately? Or should $\mathcal{M}_1$ be selected, since it is better for prediction of future data? As our third contribution we argue that a third option, which we call the *Switch-Point model*, should be considered within the regular MDL framework. The Switch-Point model represents the refined hypothesis that both $\mathcal{M}_0$ and $\mathcal{M}_1$ have merit, but for different amounts of data. We justify by the MDL principle that the Switch-Point model should be considered. In addition, we give an interpretation of the regular MDL procedure for model selection as an instance of *Bayes factors* model selection. Therefore, just like in prediction, our results transfer to the Bayesian perspective directly and the Switch-Point model should be considered by Bayesians as well.

Outline

This thesis contains four chapters. Chapter 1 serves as an introductory chapter to required background theory. It introduces the MDL and Bayesian procedures for prediction and model selection. In Chapter 2 we introduce the momentum phenomenon and Switch-Point procedure, and tentatively demonstrate that the momentum phenomenon can become a momentum problem. In Chapter 3 we prove that the momentum phenomenon may get very large as measured, roughly speaking, by the maximum improvement in predictive accuracy that any procedure might hope to gain compared to regular MDL by dealing optimally with the momentum phenomenon. In addition, we present strong evidence that the momentum phenomenon is, in fact, a momentum problem. Chapter 4 provides further discussion of the results from the previous chapters. We offer an explanation of the momentum

problem and argue that it transfers from prediction to model selection. In addition, we are able to shed more light on results in prior work by explaining them in terms of the momentum problem. Finally, we make several detailed suggestions for future investigation.

Personal Motivation

I think there are two fundamental problems in Artificial Intelligence (AI). The first is to construct a useful representation of the world; the second is to use this representation to act successfully. The definition of success, of course, depends on the task at hand. Though much progress has been made since the inception of the field of AI with Turing, I consider neither of these problems to be solved to satisfactory degree.

In adherence to the motto *first things first* my main interest in AI lies with constructing representations. I view this as the task of constructing stable symbols that represent useful aspects of a noisy environment. This is equivalent to solving the *symbol grounding problem* as defined by Harnad [1990], who poses the question: "How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads?" Harnad immediately answers his own question by proposing that "[s]ymbolic representations must be grounded bottom-up in nonsymbolic representations". That is, he suggests constructing elementary stable symbols to represent useful aspects of the environment and to construct higher-level symbols recursively by combining them.

Harnad suggests using connectionist techniques to construct elementary symbols and approaches from traditional symbolic AI to combine them into higher-level symbols. As an alternative, I would propose using models (Section 1.3) to implement symbols. Models may be used to provide elementary stable symbols by modelling inputs from sensors[1], but also to construct higher-level symbols by modelling the behaviour of lower-level symbols. Thus they unify the framework for symbol construction. In addition, by their probabilistic nature models are able to account for noise in the environment. Furthermore, by the MDL principle models can be assigned a clear interpretation regardless of whether they are true in any sense. Finally, the MDL

[1] Actually, I would consider sensor inputs to be the most elementary symbols.

principle dictates that we should evaluate models by their usefulness at describing observations. This fits well with evolutionary theories on how human intelligence may have evolved.

It would seem a formidable engineering task to construct useful models for different domains. We might find consolation, however, in realising that much work has already been done. After all, this is the process more widely known as the conduction of *science*. It is mostly the elementary symbols that have traditionally been neglected. In addition, our attempts to model the world can be greatly expedited by building an appropriate cognitive toolkit. We should develop recipes of the kind: if I assume this and that about the behaviour of lower-level symbols, then such-and-such a model will describe that behaviour efficiently. Identification of the momentum problem contributes to the construction of these recipes. It shows that it may not always be assumed that a single model describes an observed phenomenon best at all numbers of observations. This is my first contribution to tackling one of the fundamental problems in AI.

Acknowledgements

I would like to express my gratitude to all who have enabled me to write this thesis. In particular, I am much indebted to Peter Grünwald for supervising me during my student internship at the Centrum voor Wiskunde en Informatica (CWI). He was always available to act as an oracle on everything related to MDL. Unlike the oracles in antiquity, however, his answers were consistently to the point and always contained useful insights. Much of his knowledge has poured into his upcoming book [Grünwald, 2007], a preliminary version of which I have consulted many times.

This thesis also owes much to Steven de Rooij, who first suggested its topic. Our frequent fruitful discussions pruned many unproductive lines of reasoning in a matter of minutes that would have taken me days to dismiss on my own. Still, at times managing this thesis would feel like an impossible juggling act. I always knew that it should be possible in principle, however, as Steven set the example by juggling with five balls simultaneously.

The groundwork for this writing was laid at the Universiteit van Amsterdam. I would like to thank Paul Vitányi, Peter van Emde Boas and Piet Rodenburg

for their willingness to sit on my thesis committee, although, to my regret, Paul will most likely not be able to attend. My thanks also to Eric-Jan Wagenmakers, who suggested several useful references, and to everyone at CWI who contributed to the constructive, stimulating and often challenging environment that is the Quantum Computing and Advanced Systems Research group.

In addition, I am especially grateful to Klara Pigmans. Her shining personality makes the hardest difficulties melt away. Finally, I would like to thank my family and friends for their continuing support.

Tim van Erven

Amsterdam
April, 2006

Contents

List of Figures

CHAPTER 1

Introduction to MDL and Bayesian Prediction

The first section of this chapter introduces the Minimum Description Length (MDL) principle, which equates learning from a set of observations with finding a short description of the observations. The MDL principle is formalised using codes, which are closely related to probability distributions. Codes, probability distributions and their relationship will be introduced in Section 1.2. Section 1.3 introduces models as possible explanations for the observations. In the presence of multiple models, the MDL principle can be applied to the tasks of prediction and model selection. The resulting procedures are presented in Section 1.4 and also given an interpretation from the perspective of Bayesian statistics.

1.1 Minimum Description Length Principle

One of the most important parts of learning is to generalise from specific observations to general descriptions. We will call a set of observations the *data*. The *Minimum Description Length* principle states that learning is to find a short description of the data. This statement is the result of a two-step argument that goes: learning is to find regularity in the data; and any regularity in the data can be used to give a shorter description of the data. The MDL principle then follows directly.

We take the first part of the argument as a task description: find as much regularity in the data as possible. The second part, the duality between regularity and short descriptions, will now be illustrated by an example adapted from [Grünwald et al., 2005]. Consider the following two sequences of coin flips by a swindler, who may have tampered with the outcomes. `H` denotes *heads* and `-` denotes *tails*:

```
---H---H---H---H---H ... ---H---H---H---H---H---H---H
-HHH-H--HH-H--H--HH- ... H-H-HHH-H-HHH-HH---H-HH---H-
```

Each of the sequences is 1000 flips long, but to save print we have only listed the start and the end of each sequence. The first sequence consists of 2500 repetitions of the part "`---H`". To write it out in full requires 1000 bits, one bit per coin flip. It might have been described much shorter in any general-purpose programming language by a program like:

```
for i in 1 to 2500; do print "---H"; end
```

Clearly this program can be described in far less than 1000 bits; it exploits the regularity in the sequence to describe it using less bits. The second sequence has been generated by tosses of a fair coin and contains no useful regularity. The shortest program to describe it will look like:

```
print "-HHH-H--HH-H--H--HH- ... H-H-HHH-H-HHH-HH---H-HH---H-"
```

This program cannot use any regularity to give a shorter description of the sequence. It is therefore of approximately the same length as the original sequence plus some minor syntactic overhead.

The exact length of the programs depends on the specifics of the programming language that is used[1]. What has here been called a programming language corresponds to what will be called a *code* in the remainder of this thesis, although codes will often be highly specialised rather than general-purpose; a program will be called a *codeword*; and the length of a unique program to generate a specific data sequence will be referred to as the *codelength* for the data. All results in this thesis concern the relative merit and construction of codes. Codes will be compared by the codelength they assign to the data. Codes, codelengths and related concepts will be formally introduced in the next section.

[1] We view a programming language as a Turing machine, a program as its input and the data as its output.

1.2 Codes and Probability Distributions

Codelength functions are closely related to probability distributions. Before formalising the concept of codes, we therefore first introduce probability distributions. The relationship between codelength functions and distributions depends on the *Kraft inequality*, which we will state next. We will then give an interpretation of codelength as accumulated *predictive loss* that will be used throughout this thesis and finally conclude this section by introducing *universal codes*, which arguably may be considered the most important concept in MDL.

1.2.1 Probability Distributions

A probability distribution is a mathematical construct that satisfies certain formal properties. It is often interpreted as a statement about the relative frequency of the outcomes of a nondeterministic experiment if the exact same experiment were repeated infinitely many times. The relative frequency assigned to a specific outcome is called the probability of that outcome. We will now state the most important formal properties of probability distributions, using the related notions of *events* and *outcome spaces*.

An event is a set of possible outcomes of a nondeterministic process. The set of all possible outcomes, $\mathcal{X}$, is called the outcome space. Events therefore correspond to subsets of $\mathcal{X}$. Let $\mathcal{P}(\mathcal{X})$ denote the power set of $\mathcal{X}$. Then a probability distribution $P : \mathcal{P}(\mathcal{X}) \to [0, 1]$ maps events to their probability. The probability that some outcome will occur must always be one, i.e. $P(\mathcal{X}) = 1$, and the probability that no outcome will occur must be zero, i.e. $P(\emptyset) = 0$. In addition, the probability of any two disjoint events equals their joint probability. That is, for any two events $\mathcal{E}_1$ and $\mathcal{E}_2$ such that $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$, it holds that $P(\mathcal{E}_1 \cup \mathcal{E}_2) = P(\mathcal{E}_1) + P(\mathcal{E}_2)$.

An example of a probability distribution is the distribution that considers all outcomes equally likely. This distribution is called the *uniform distribution*. It assigns the same probability $P(x) = 1/|\mathcal{X}|$ to each outcome $x \in \mathcal{X}$.

If it is known that the outcome of an experiment will be one of the outcomes in event $\mathcal{E}$, then the probability of any outcome that is not in $\mathcal{E}$ must be zero and the probability of the other outcomes is normalised such that the

total probability sums to one again. The resulting probability distribution is called conditional on $\mathcal{E}$ and the probability that it assigns to any event $\mathcal{E}'$ is denoted by $P(\mathcal{E}'|\mathcal{E})$. Thus conditional probability is defined as

$$P(\mathcal{E}'|\mathcal{E}) := \frac{P(\mathcal{E}' \cap \mathcal{E})}{P(\mathcal{E})}. \tag{1.1}$$

Frequently, events can conveniently be described using *random variables*. A random variable $X : \mathcal{X} \to \mathbb{R}$ maps individual outcomes to the real numbers. A value x for X therefore corresponds to an event. If $\mathcal{E}$ denotes this event, then we may write $P(X = x)$ equivalently with $P(\mathcal{E})$. When no confusion is possible we abbreviate $P(X = x)$ to $P(x)$.

Data

A series of n observations that constitute the data may more formally be considered outcomes from a series of n experiments. We will call the number of outcomes n the *sample size*. Suppose we would like to describe n observations, then we might consider outcomes $x_1, \ldots, x_n$ from outcome spaces $\mathcal{X}_1, \ldots, \mathcal{X}_n$ or, equivalently, a single composite outcome $x^n := (x_1, \ldots, x_n)$ from the joint outcome space $\mathcal{X}^n := \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$. Suppose further that we would like to define a probability distribution for any sample size. Then we may consider a sequence of related probability distributions $P^1, P^2, \ldots$ on outcome spaces $\mathcal{X}^1, \mathcal{X}^2, \ldots$. Any two distributions P^n and P^{n+1} are called *compatible* if the marginal distribution for P^{n+1} restricted to n outcomes is equal to P^n. That is, for all $x^n \in \mathcal{X}^n$ it needs to hold that $P^n(x^n) = \sum_{x_{i+n} \in \mathcal{X}_{n+1}} P^{n+1}(x^n, x_{n+1})$. If every two consecutive distributions in the sequence are compatible, then the sequence of probability distributions is called a *(probabilistic) source* [Grünwald et al., 2005]. Whenever the sample size n is clear from context we will write $P(x^n)$ instead of $P^n(x^n)$.

Any probabilistic source defines a unique infinite sequence of conditional distributions $P_1(x_1)$, $P_2(x_2|x^1)$, …, $P_n(x_n|x^{n-1})$, … by

$$P_n(x_n|x^{n-1}) := \frac{P^n(x^n)}{P^{n-1}(x^{n-1})}. \tag{1.2}$$

It follows by compatibility of P^n and P^{n-1} that the probabilities of P_n sum to one: $\sum_{x_{n+1}} P_n(x_{n+1}|x^n) = 1$. Conversely, any such sequence defines a unique

probabilistic source by repeated application of the definition of conditional probability:

$$P^n(x^n) = \prod_{i=1}^{n} P_i(x_i|x^{i-1}). \tag{1.3}$$

Compatibility is verified by

$$\begin{aligned}
\sum_{x_{n+1}} P^{n+1}(x^{n+1}) &= \sum_{x_{n+1}} \prod_{i=1}^{n+1} P_i(x_i|x^{i-1}) \\
&= \prod_{i=1}^{n} P_i(x_i|x^{i-1}) \sum_{x_{n+1}} P_{n+1}(x_{n+1}|x^n) \\
&= \prod_{i=1}^{n} P_i(x_i|x^{i-1}) \\
&= P^n(x^n),
\end{aligned}$$

which shows that these distributions form a source.

Suppose that the data, x^n, are distributed according to P^n in source P. Then we say that the data have been *sampled from* P and that P is the *generating source* for the data. The data may be sampled sequentially. For instance, if we sample consecutive outcomes x_1, x_2, $\ldots$, x_n in a time-series according to $P_1(x_1)$, $P_2(x_2|x^1)$, $\ldots$, $P_n(x_n|x^{n-1})$, then by Equation (1.2) the joint outcomes x^n are distributed according to P^n.

Empirical Distribution

Given a series of n observations $x^n \in \mathcal{X}^n$ we may construct the *empirical distribution* $\mathbb{P}_{x^n}$ of x^n over outcomes $y \in \mathcal{X}$, which sets the probability of observing any y equal to the relative frequency of y in x^n. $\mathbb{P}_{x^n}$ is an element of the unit simplex Δ^m in $\mathbb{R}^m$ and is defined as

$$\mathbb{P}_{x^n}(y) := \frac{n_y(x^n)}{n},$$

where $n_y(x^n)$ denotes the number of occurrences of y in x^n. The unit simplex in $\mathbb{R}^m$ is defined by

$$\Delta^m := \left\{ (y_1, \ldots, y_m)^T \in \mathbb{R}^m : \sum_{i=1}^{m} y_i = 1, y_1, \ldots, y_m \geq 0 \right\}.$$

Expectation

We will frequently be interested in typical values of some function $f(X)$ of a random variable X that is distributed according to distribution P. A common way to summarise the value of $f(X)$ on typical values of X is by the *expected* value of $f(X)$ under P, which is defined by

$$E_{X\sim P}[f(X)] := \sum_{x\in\mathcal{X}} P(x)\cdot f(x)$$

and may be interpreted as a weighted average of the value of $f(X)$ according to P. We will always abbreviate the subscript to P whenever X is clear from context.

1.2.2 Codes

The Minimum Description Length principle concerns the length of descriptions. We capture the notion of descriptions using codes and codelengths. A code may be interpreted as a strategy for transmitting the data to another party over a binary communication channel. Formally, a code $C : \mathcal{X} \to \mathcal{D}^*$ for random variable X is defined as a one-to-one mapping from $\mathcal{X}$ to $\mathcal{D}^*$, where $\mathcal{X}$ is the outcome space for X and $\mathcal{D}^*$ is the set of finite strings of symbols from the alphabet $\mathcal{D}$. We use $C(x)$ to denote the *codeword* assigned by code C to outcome $x \in \mathcal{X}$. Each codeword is a string in $\mathcal{D}^*$. An alphabet $\mathcal{D}$ of size D is called D-ary. We will assume throughout this thesis that $\mathcal{D}$ is binary (2-ary). Without loss of generality we furthermore assume that $\mathcal{D} = \{0, 1\}$.

To connect to the terminology from Section 1.1 we note that outcomes of the random variable X correspond to the sequences of coin flips in the example of that section. If the data consist of n observations we will also let $x^n := x_1, \ldots, x_n$ denote outcomes from a compound random variable that will be written X^n. Individual observations will then be called outcomes. n is one thousand for the sequences in the previous example. What is here called a code corresponds to what has previously been called a programming language, except that we require each possible sequence of outcomes to have only a single corresponding program. Codewords correspond to the previous programs.

In this thesis we restrict ourselves to *prefix codes*, which are codes such that no codeword is a prefix of any other codeword [Cover and Thomas, 1991]. The codewords for prefix codes are self-delimiting. This enables us to code a sequence of outcomes by concatenating the codewords for the individual outcomes without the need for any special symbol to separate codewords. In the following, we will always use the term code to refer to a prefix code.

A function L_C that maps outcomes x to the length of their codeword $C(x)$, will be called a *codelength function* and we will call $L_C(x)$ the *codelength* assigned to x by code C. When C is clear from context, we sometimes omit the subscript and write $L(x)$ instead of $L_C(x)$. The MDL principle implies that we should compare codes by the codelength they assign to the data. In fact, it is *only* interested in codelength. The actual codes, as long as they exist, are of no importance. From the MDL perspective a code is obviously inefficient if there exists a code that assigns shorter codelength to at least one outcome and the same codelength to all other outcomes. A code that is not obviously inefficient in this way is called *complete*. To be precise, a code C is complete if no other code C' exists such that for all $x \in \mathcal{X}$, $L_{C'}(x) \leq L_C(x)$ while for at least one $x \in \mathcal{X}$, $L_{C'}(x) < L_C(x)$ [Grünwald et al., 2005].

Fixed-Length Codes

A code that assigns the same codelength $\log |\mathcal{X}|$ — we always take the logarithm to base 2 — to all outcomes is called a *fixed-length code*. For instance, for outcome space $\mathcal{X} = \{a, b, c, d\}$, $|\mathcal{X}| = 4$, the following code C would be fixed-length for outcomes $x \in \mathcal{X}$:

x	a	b	c	d
$C(x)$	00	01	10	11

Fixed-length codes are worst-case optimal in the sense that all non-fixed-length codes assign longer codewords to at least one outcome. To see this for the previous example consider a non-fixed-length code that would assign codeword 0 to outcome a. Then by the prefix condition on codes, this code would be prevented from using codewords 00 and 01. It would therefore have to use codewords starting with 1 to code outcomes b, c and d. If it were to use 1 itself as a codeword, then by the prefix condition it would be prevented from assigning any other codewords. Therefore 1 cannot be a codeword in

this code. But then only two codewords starting with 1 of length two remain to code a, b and c. Therefore by necessity at least one of these outcomes would have to be assigned a codeword of length at least 3, which is longer than its codeword in the fixed-length code above.

1.2.3 Kraft Inequality

Codelength functions and probability distributions are closely related: for any probability distribution P a so-called *Shannon-Fano code* with codelengths $L(x_i)$ can be constructed such that $L(x_i) = \lceil -\log P(x_i) \rceil$ and vice versa. The codelengths are rounded up, because they are restricted to integer values by definition. No rounding is required when going from codelengths to probabilities. The exact correspondence between codes and probability distributions is expressed by the Kraft inequality [Cover and Thomas, 1991], which states:

Theorem 1.2.1 *For any finite number of codewords $C(x_1)$, ..., $C(x_m)$ that form a code, the corresponding codelengths $L(x_1)$, ..., $L(x_m)$ must satisfy the inequality*

$$\sum_{i=1}^{m} 2^{-L(x_i)} \leq 1.$$

Conversely, given any set of codelengths $L(x_1), L(x_2), \ldots, L(x_m)$ that satisfy this inequality, there exists a code with these codelengths.

The probabilities $P(x_i)$ for any probability distribution P sum to one. Therefore the set of codelengths $L(x_i) = \lceil -\log P(x_i) \rceil$ satisfy the Kraft inequality:

$$\sum_{i=1}^{m} 2^{-\lceil -\log P(x_i) \rceil} \leq \sum_{i=1}^{m} 2^{\log P(x_i)} = \sum_{i=1}^{m} P(x_i) = 1.$$

It follows that a code with codelengths $L(x_i) = \lceil -\log P(x_i) \rceil$ can be constructed for any P.

Conversely, for any code with codelengths $L(x_i)$ a corresponding probability distribution P can be constructed with probabilities $P(x_i) = 2^{-L(x_i)}$. The Kraft inequality then ensures that the sum of the probabilities $P(x_i)$ will not exceed one. If the sum of the probabilities of P is less than one, then

P is called *defective*. A defective distribution may be viewed as an ordinary distribution that assigns some of its probability to an imaginary outcome $\tilde{a}$ that will never be observed. Defective distributions are related to inefficiency in codes. To be precise, it can be proved that a code with codelengths $L(x_i)$ is complete if and only if the corresponding distribution with probabilities $P(x_i) = 2^{-L(x_i)}$ is not defective [Grünwald, 2007].

In the discussion above codelength functions and probability distributions are not placed on the same footing. If we start with codelengths $L(x_i)$, then an exactly corresponding distribution with probabilities $P(x_i) = 2^{-L(x_i)}$ can always be constructed; however, if we start with a distribution, then the corresponding codelengths $-\log P(x_i)$ must be rounded up to the nearest integer. This asymmetry can be removed by dropping the integer requirement for codelengths.

1.2.4 Non-integer Codelengths

To get an exact correspondence between probabilities and codelengths, we would like to remove the requirement that codelengths be integers. As argued extensively by Grünwald [2007], who calls non-integer codelengths *idealised codelengths*, removing this requirement has many advantages and very few disadvantages. In the remainder of this thesis we will therefore drop the integer requirement for codelengths and use idealised codelengths throughout. For completeness we will now present three arguments to support our position. The first two are from [Grünwald, 2007]. The third is based on [Dawid, 1992b].

Rounding has Negligible Influence on Codelength

Consider a code with codelengths $L(x_i) = \lceil -\log P(x_i) \rceil$ for outcomes $x_i \in \mathcal{X}_i$. Then the prefix property guarantees that multiple outcomes in a sequence can be coded by concatenating codewords: $L(x^n) = \sum_{i=1}^{n} L(x_i)$. It would appear at first sight that the increase in codelength due to rounding accumulates. We might think that the average codelength per outcome is given by

$$\frac{L(x^n)}{n} = \frac{\sum_{i=1}^{n} \lceil -\log P(x_i) \rceil}{n}.$$

However, the rounding can be spread out over all outcomes by viewing x^n as a single outcome from the outcome space $\mathcal{X}^n$. Then applying the Kraft inequality to $L(x^n)$ gives

$$\frac{L(x^n)}{n} = \frac{\lceil -\log \prod P(x_i) \rceil}{n} < \frac{1}{n} + \frac{\sum -\log P(x_i)}{n}.$$

The influence of rounding on the average codelength per outcome decreases rapidly with the length of the sequence.

Invariance to Alphabet Size

Hitherto we have assumed binary codewords. We might however with equal validity have chosen a ternary alphabet, or any D-ary alphabet for that matter, taking the logarithm not to base 2 but to base D where applicable. It is easy to see that the effect of rounding can be very different depending on the alphabet size D that is chosen. Compare for instance an outcome a with probability $P(a) = \frac{1}{8}$. In the binary alphabet rounding would not influence the outcome and its codelength would be $L_2(a) = \lceil -\log(\frac{1}{8}) \rceil = \lceil 3 \rceil = 3$. In the ternary alphabet, however, its codelength would be $L_3(a) = \lceil \log_3(\frac{1}{8}) \rceil \approx \lceil 1.89 \rceil = 2$, a rounding-off effect of nearly 0.11 ternary bits.

Suppose we remove the restriction to integer codelengths. Then for all possible outcomes a the ratio between codelengths under distinct alphabet sizes D_1 and D_2 is the same and depends only on the sizes of the alphabets:

$$\frac{L_{D_1}(a)}{L_{D_2}(a)} = \frac{-\log_{D_1} P(a)}{-\log_{D_2} P(a)} = \frac{-\ln P(a)/\ln D_1}{-\ln P(a)/\ln D_2} = \frac{\ln D_2}{\ln D_1}.$$

With the integer requirement, rounding-off effects may be different for different alphabet sizes and no such invariant relationship holds.

Distributions are their own Optimal Codes

Suppose the data are generated by sampling from a distribution P. That is, we assume that they are generated by a procedure that would generate each sequence of observations x^n with relative frequency going to $P(x^n)$ if it were repeated infinitely many times. Alternatively, and perhaps more appropriately from an MDL perspective, we may view sampling from P as

assigning relative importance $P(x^n)$ to any sequence x^n. Then the code C with codelengths $L_C(X) = -\log P(X)$ is optimal to code the data in the sense that it will in expectation and with high probability assign shortest codelength to the data among all possible codes. To be precise, the expected codelength

$$E_P[L_C(X)] := \sum_{x \in \mathcal{X}} P(x) \cdot L_C(x) \tag{1.4}$$

is minimised if $L_C(X) = -\log P(X)$ [Dawid, 1992b]; and the probability that any other code C' achieves significantly shorter codelength is bounded by

$$P(-\log P(X) \geq L_{C'}(X) + c) \leq 2^{-c} \quad \text{with } c > 0, \tag{1.5}$$

which is exponentially small in the number of bits c that the alternative code C' should gain [Dawid, 1992b]. The optimality for P of the code with codelengths $-\log P(X)$ strongly ties the two together. This is another reason to drop the requirement that codelengths be restricted to the integers.

1.2.5 Kullback-Leibler Divergence

Suppose the data X are distributed according to P_1. Then the expected number of additional bits needed when coding the data with a code corresponding to distribution P_2 is called the *Kullback-Leibler divergence*. The Kullback-Leibler divergence from P_1 to P_2 is defined as

$$\begin{aligned} D(P_1 \| P_2) &:= E_{P_1}\left[\log \frac{P_1(X)}{P_2(X)}\right] \\ &= E_{P_1}\left[-\log P_2(X) - [-\log P_1(X)]\right]. \end{aligned}$$

Based on continuity arguments, we use the convention that $0 \log \frac{0}{P_2(x)} = 0$ and $P_1(x) \log \frac{P_1(x)}{0} = \infty$ for $P_1(x) \neq 0$ [Cover and Thomas, 1991]. It can be shown that $D(P_1 \| P_2) \geq 0$ and that $D(P_1 \| P_2) = 0$ if and only if $P_1 = P_2$. Kullback-Leibler divergence admits several different interpretations in information theory and statistics [Clarke and Barron, 1990]. For instance, it is sometimes interpreted as a measure of the difference between P_1 and P_2.

1.2.6 Codelength as Accumulated Predictive Loss

Data that arrive sequentially at intervals are called *time-series* data. Suppose we observe a sequence of data $x_1, x_2, \ldots, x_n$ in a time-series and after

each new observation our task is to predict the next observation. In other words, at each time-step n we have to predict x_{n+1} given all previous observations $x_1, \ldots, x_n = x^n$. Let us express our strategy for prediction as conditional probability distributions $P(\cdot|x^n)$ over $\mathcal{X}_{n+1}$, the outcome space for x_{n+1}. Then a common way to measure the success of our consecutive predictions is by their *log loss* [e.g. Good, 1952]. The log loss also arises as a natural measure of success in different settings such as sequential gambling [Kelly, Jr., 1956; Barron, 1998]. The log loss of prediction $P(\cdot|x^n)$ is given by $-\log P(x_{n+1}|x^n)$ if x_{n+1} is the next observation that actually occurs. We will refer to the log loss of a distribution that predicts the next outcome as predictive (log) loss.

We introduce $C(x_{n+1}|x^n)$ as new notation to denote a conditional code for outcome x_{n+1} that depends on the previous outcomes x^n and let the corresponding codelengths be denoted by $L_C(x_{n+1}|x^n)$.

The predictive log loss equals the codelength of a code with conditional codelengths $L_C(x_{n+1}|x^n) = -\log P(x_{n+1}|x^n)$. By repeated application of Equation (1.1) the accumulated predictive loss over the first n observations turns out to be equal to the total codelength of a code C' over n outcomes, where C' is completely defined by the predictive strategy being used:

$$\begin{aligned}\sum_{i=1}^{n} -\log P(x_{i+1}|x^i) &= -\log \prod_{i=1}^{n} P(x_{i+1}|x^t) \\ &= -\log P'(x^n) \\ \sum_{i=1}^{n} L_C(x_{i+1}|x^i) &= L_{C'}(x^n). \end{aligned} \tag{1.6}$$

We call Equation (1.6) the *sequential decomposition of codelength*. By the equality between conditional codelength and predictive loss, it implies that the codelength of a sequence of n observations can be viewed as the accumulated log loss of predicting these observations in a time-series.

In addition to the sequential decomposition of codelength for individual sequences we will now define the *sequential decomposition of expected codelength*. Suppose the data are generated by sampling successive outcomes $x_1, x_2, \ldots$ from a sequence of probability distributions $P_1, P_2, \ldots$, where each outcome x_i is sampled according to P_i and each P_i is allowed to depend on

the outcomes generated by its predecessors $P_1, \ldots, P_{i-1}$ in the sequence. This is the setting of all examples that will be presented in this thesis. Then

$$\begin{aligned} E_{P_1,\ldots,P_n}[L_C(X^n)] &= E_{P_1,\ldots,P_n}\left[\sum_{i=1}^{n} L_C(X_i|X^{i-1})\right] \\ &= \sum_{i=1}^{n} E_{P_1,\ldots,P_n}[L_C(X_i|X^{i-1})] \\ &= \sum_{i=1}^{n} E_{P_1,\ldots,P_i}[L_C(X_i|X^{i-1})]. \end{aligned} \tag{1.7}$$

We call Equation (1.7) the sequential decomposition of expected codelength. By the equality between conditional codelength and predictive loss, it implies that the expected codelength of a sequence of n observations can be viewed as the accumulated expected log loss of predicting these observations in a time-series.

By the equivalence between the (expected) codelength of a sequence of observations and the (expected) accumulated log loss of sequentially predicting them, it is justified to use these descriptions interchangeably. In the remainder of this thesis we will therefore freely switch between them whenever it facilitates our presentation.

1.2.7 Universal Codes

Suppose we are given any countable set of codes $\mathcal{C} = \{C_1, C_2, \ldots\}$ with codelength functions that correspond to distributions $\mathcal{P} = \{P_1, P_2, \ldots\}$. Then it is possible to construct a so called *universal code* $C_{\mathcal{C}}$ for $\mathcal{C}$ that assigns nearly as short codelength to any outcome x as the code in $\mathcal{C}$ that assigns shortest codelength to x. The probability distributions that correspond to universal codes are called *universal models*. The use of universal models may be considered the main characterising feature of the MDL principle [Grünwald et al., 2005]. There exist many types of universal models. In this thesis we frequently use so-called *Bayesian universal models* to construct *Bayesian universal codes*. A Bayesian universal model $P_{\mathcal{P}}$ is a mixture of the distributions in $\mathcal{P}$ according to some distribution $w_{\mathcal{P}}$ over the elements of $\mathcal{P}$ that assigns positive probability to all of them. $P_{\mathcal{P}}$ is defined by

$$P_{\mathcal{P}}(x) := \sum_i w_{\mathcal{P}}(P_i) P_i(x). \tag{1.8}$$

It can be verified that $P_{\mathcal{P}}$ assigns probability one to the entire outcome space by

$$\begin{aligned}\sum_x P_{\mathcal{P}}(x) &= \sum_x \sum_{i=1}^{m} w_{\mathcal{C}}(P_i) P_i(x) \\ &= \sum_{i=1}^{m} w_{\mathcal{C}}(P_i) \sum_x P_i(x) \\ &= \sum_{i=1}^{m} w_{\mathcal{C}}(P_i) \\ &= 1.\end{aligned}$$

By the Kraft inequality it is possible to construct $C_{\mathcal{C}}$ such that $L_{C_{\mathcal{C}}}(x) = -\log P_{\mathcal{P}}(x)$ for any outcome x. Now the difference in codelength between $C_{\mathcal{C}}$ and the best code C_a in $\mathcal{C}$ can be bounded by:

$$\begin{aligned}L_{C_{\mathcal{C}}}(x) &= -\log P_{\mathcal{P}}(x) \\ &= -\log \sum_i w_{\mathcal{P}}(P_i) P_i(x) \\ &\leq -\log w_{\mathcal{P}}(P_a) P_a(x) \\ &= -\log w_{\mathcal{P}}(P_a) + L_{C_a}(x).\end{aligned}$$

Therefore $C_{\mathcal{C}}$ assigns at most $-\log w(P_a)$ more bits to the data than the best code C_a in $\mathcal{C}$. We will frequently design codes for n consecutive outcomes $x^n = x_1, \ldots, x_n$. In this case the codelength $L_{C_a}(x^n)$ will be large for all x^n. The constant $-\log w(P_a)$ bits overhead for $C_{\mathcal{C}}$, which do not depend on n, will then be negligible relative to $L_{C_a}(x^n)$.

In many cases $C_{\mathcal{C}}$ is not in $\mathcal{C}$. We therefore might wonder whether it ever assigns shorter codelength to any outcome x than the best code for x in $\mathcal{C}$. However, by

$$\begin{aligned}L_{C_{\mathcal{C}}}(x) &= -\log \sum_i w_{\mathcal{P}}(P_i) P_i(x) \\ &\geq -\log \sum_i w_{\mathcal{P}}(P_i) \max_a P_a(x) \\ &= -\log \max_a P_a(x) \\ &= \min_a -\log P_a(x)\end{aligned}$$

it is proved that this is not possible. We conclude that the codelength $L_{C_\mathcal{C}}(x)$ behaves *approximately* like the codelength of the best code in $\mathcal{C}$.

For uncountable sets $\mathcal{C}$, Bayesian universal codes can be constructed in a similar way to the countable case. However, the additional codelength they assign to the data compared to the best code in $\mathcal{C}$ cannot be bounded by a constant that does not depend on n. Under regularity conditions on the sets, however, it can be bounded by $O(\log n)$. As $L_{C_a}(x^n)$ will usually grow linearly in n, for large n this may still be considered relatively small.

In some cases we may prefer a universal code that explicitly identifies a single best element P_a in $\mathcal{P}$. For countable $\mathcal{P}$ this is realised by *two-part codes*. Given a distribution $w_\mathcal{P}$ over $\mathcal{P}$, a two-part code first codes the index of P_a in $-\log w_\mathcal{P}(P_a)$ bits — its first part — and then codes the data using P_a in $-\log P_a(x^n)$ bits — its second part. Thus its codelength is given by

$$L(x) = \min_a \left\{ -\log w_\mathcal{P}(P_a) + L_{C_a}(x) \right\}.$$

Note that the choice of $w_\mathcal{P}$ biases the code towards certain P_a. Two-part codes are less efficient than Bayesian universal codes. If explicit identification of P_a is not required, we therefore prefer Bayesian universal codes.

1.3 Models

It is common in science to consider multiple explanations for the same phenomenon. We call each explanation $\mathcal{M} \in \mathbb{M}$ a *model*. There appears to be no standard terminology for the set $\mathbb{M}$ of models. We will call $\mathbb{M}$ the *model set*. Models are formalised as (possibly uncountably infinite) sets of probabilistic sources over the same (joint) outcome spaces. In particular we restrict ourselves to *parametric models*

$$\mathcal{M} := \{P_\theta : \theta \in \Theta \subseteq \mathbb{R}^d\},$$

which are models $\mathcal{M}$ such that each probabilistic source P_θ in the model is indexed by a unique parameter θ in d-dimensional *parameter space* Θ. We assume that the model set $\mathbb{M}$ consists of countably many models containing sources over the same outcome spaces. In all situations considered in this thesis the model set will contain at most three models.

1.3.1 Bernoulli Models

To illustrate the notion of a model, we will now present two models that contain Bernoulli distributions. The Bernoulli distribution on single binary outcomes x is defined as

$$P_\theta(x) := \theta^x \cdot (1-\theta)^{1-x} \quad \text{with } x \in \mathcal{X} = \{0,1\},$$

where θ defines the probability of observing a one: $P_\theta(X = 1) = \theta$. If a single distribution is extended to multiple outcomes by taking product distributions, then the resulting sequence of distributions is always compatible. A *Bernoulli probabilistic source* can therefore be defined by taking product distributions:

$$P_\theta(x^n) := \theta^{n_1(x^n)} \cdot (1-\theta)^{n_0(x^n)},$$

where $n_1(x^n)$ and $n_0(x^n)$ denote the number of ones and the number of zeroes in x^n respectively. It holds for all x^n that $n_1(x^n) + n_0(x^n) = n$.

Our first example is the *Bernoulli model*, which is one of the most well-known models. We define it as

$$\mathcal{M}_1 := \{P_\theta : \theta \in \Theta = (0,1)\},$$

where P_θ is the Bernoulli source based on the Bernoulli distribution with $P_\theta(X = 1) = \theta$. It should be noted that the Bernoulli model is usually defined with $\Theta = [0,1]$. In our modified definition the Bernoulli model is an *exponential family* whereas in the common definition it is not. The exponential families are a group of models that share many useful regularity properties [Barndorff-Nielsen, 1978; Grünwald, 2007]. They can be parameterised in a certain canonical form. To discuss them more extensively lies outside the scope of this thesis. We will therefore only state their regularity properties without discussion when required, together with a reference to more information.

Our second example of a model contains only a single Bernoulli source and does not have a standard name. We will refer to it as the *Single Bernoulli model*. It is defined as

$$\mathcal{M}_0 := \{P_{0.6}\},$$

where $P_{0.6}$ is the Bernoulli source P_θ with $\theta = 0.6$. We will use both Bernoulli models in our examples. The choice of $P_{0.6}$ in the Single Bernoulli model is arbitrary to the extent that we wished to avoid special distributions — such as the symmetric distribution $P_{0.5}$, for instance —, which might have introduced more regularity than desired into our results.

1.4 MDL and Bayes

The MDL principle can be applied to the problem of predicting the next unobserved outcome x_{n+1} given data x^n in the presence of multiple models. In addition it can be applied to select a single model that may be considered the best explanation for x^n among all models in the model set. The former we will call the task of *prediction*, the latter the task of *model selection*. Model selection is sometimes also called hypothesis testing [Leamer, 1978]. In this thesis we will focus on prediction. However, as the tasks are closely related [Chickering and Heckerman, 2000], our results also have implications for model selection. We will point these out in Section 4.3.

Applying MDL to prediction or model selection amounts to constructing a single appropriate code to describe the data. We will compare standard choices for this code to an alternative called the Switch-Point procedure, which will be presented in Section 2.2. For these standard choices the resulting procedures can also be given an interpretation from the so-called *Bayesian* perspective. To improve upon them therefore means to improve on the Bayesian approach as well. For the convenience of Bayesian readers we include a brief introduction to the Bayesian approach to prediction and model selection and will try to point out when our results carry over. For ease of presentation we will now first introduce the Bayesian view on prediction and model selection; then we will do the same for MDL; and finally we will connect the resulting procedures. We would like to emphasise at this point that MDL and Bayesian reasoning are very different in spirit and certainly do not *always* lead to the same procedures.

1.4.1 Bayes

Given data x^n, each probabilistic source $P_\theta \in \mathcal{M}$ defines a probability distribution P_θ^n on the data. From one Bayesian point of view each P_θ^n in a model represents a possible *explanation* for the data. A model in this view is a mixture of the distributions $\{P_\theta^n : P_\theta \in \mathcal{M}\}$ weighted by a prior distribution w over θ that does not depend on n. Thus each model defines a probability distribution

$$P_{\mathcal{M}}^n(x^n) = \int_{\theta \in \Theta} P_\theta(x^n) w(\theta)\, d\theta \tag{1.9}$$

over the data. Compatibility of the universal models $P^1_{\mathcal{M}}, P^2_{\mathcal{M}}, \ldots$ for different sample sizes follows by compatibility of the individual sources in the model:

$$\begin{aligned}\sum_{x_{n+1}} P^{n+1}_{\mathcal{M}}(x^{n+1}) &= \sum_{x_{n+1}} \int_{\theta\in\Theta} P^{n+1}_{\theta}(x^{n+1}) w(\theta)\ d\theta \\ &= \int_{\theta\in\Theta} \sum_{x_{n+1}} P^{n+1}_{\theta}(x^{n+1}) w(\theta)\ d\theta \\ &= \int_{\theta\in\Theta} P^{n}_{\theta}(x^{n}) w(\theta)\ d\theta \\ &= P^n_{\mathcal{M}}(x^n). \end{aligned} \tag{1.10}$$

Therefore $P_{\mathcal{M}}$ is a probabilistic source.

From an MDL perspective, each distribution $P^n_{\mathcal{M}}$ is a Bayesian universal model (see Section 1.2.7) for the codes corresponding to the distributions P^n_θ. In fact, Bayesian universal models derive their name from their application in Bayesian statistics. As we will see in Section 1.4.2, however, their usefulness is not limited to Bayesian statistics.

Subjective Bayesian Priors

$w(\theta)$ may be interpreted from a *subjective Bayesian* point of view as the subjective degree of belief that a probabilistic source P_θ represents the true source for the data. In the subjective view different researchers may use different w depending on their past experiences and personal beliefs. In many practical situations, however, detailed knowledge of degrees of belief about the source for the data is unavailable, for instance because it cannot feasibly be obtained from domain experts. Or worse, it may be known that none of the probabilistic sources in the model corresponds to the real source for the data as is the case, for example, when hidden Markov models are used to model human speech in speech recognition. Then the degree of belief in each probabilistic source is zero! In these cases w can still be given an *objective Bayesian* interpretation as a pragmatic choice that aims for the Bayesian methods to perform well on a given task [Berger and Pericchi, 2001].

Objective Bayesian Priors

From a subjective Bayesian point of view the prior w is fully specified by existing prior belief. In the objective Bayesian view, however, there exists no obvious default choice. In this case *Jeffreys' prior* [Grünwald et al., 2005] may be used, which has a number of appealing properties that might be expected from a prior representing ignorance. For instance, it is invariant under reparameterisation of the model. Jeffreys' prior is defined as

$$w_{\text{Jeffreys}}(\theta) := \frac{\sqrt{|I(\theta)|}}{\int_{\theta\in\Theta} \sqrt{|I(\theta)|}\, d\theta},$$

where $I(\theta)$ represents the *Fisher information matrix* and $|I(\theta)|$ denotes its determinant. The (i,j)-th element of $I(\theta)$ is given by

$$I_{ij}(\theta) := \lim_{n\to\infty} \frac{1}{n} E_\theta \left[-\frac{\partial^2}{\partial\theta'_i \partial\theta'_j} \ln P_{\theta'}(X^n) \right]_{\theta'=\theta} \tag{1.11}$$

for $i,j = 1,\ldots,d$ where d denotes the dimensionality of the parameter space. If a model contains only a single source P_θ, then the Fisher information matrix is undefined. In this case we will assume that $w_{\text{Jeffreys}}(\theta) = 1$, which is the only possible prior in that case.

Prediction

The task of prediction requires to predict the next outcome x_{n+1} given data x^n. Let P^n_{Bayes} denote the Bayesian universal model for n outcomes given some prior $w_{\mathbb{M}}$ over the models in the model set. Then the sequence $P_{\text{Bayes}} = P^1_{\text{Bayes}}, P^2_{\text{Bayes}}, \cdots$ forms a probabilistic source by the same reasoning as in (1.10). From a Bayesian perspective the next outcome should now be predicted according to the conditional distribution $P_{\text{Bayes}}(x_{n+1}|x^n)$. By (1.8) and (1.2) it follows that $P_{\text{Bayes}}(x_{n+1}|x^n)$ is given by

$$\begin{aligned} P_{\text{Bayes}}(x_{n+1}|x^n) &= \frac{P_{\text{Bayes}}(x^{n+1})}{P_{\text{Bayes}}(x^n)} \\ &= \frac{\sum_{\mathcal{M}\in\mathbb{M}} P_{\mathcal{M}}(x^{n+1}) w_{\mathbb{M}}(\mathcal{M})}{\sum_{\mathcal{M}'\in\mathbb{M}} P_{\mathcal{M}'}(x^n) w_{\mathbb{M}}(\mathcal{M}')} \\ &= \sum_{\mathcal{M}\in\mathbb{M}} P_{\mathcal{M}}(x_{n+1}|x^n) \frac{P_{\mathcal{M}}(x^n) w_{\mathbb{M}}(\mathcal{M})}{\sum_{\mathcal{M}'\in\mathbb{M}} P_{\mathcal{M}'}(x^n) w_{\mathbb{M}}(\mathcal{M}')}, \end{aligned} \tag{1.12}$$

where $P_{\mathcal{M}}$ has been defined in (1.9) and is itself a weighted average. The conditional distributions $P_{\mathcal{M}}(x_{n+1}|x^n)$ are computed by (1.2). Prediction according to $P_{\text{Bayes}}(x_{n+1}|x^n)$ is called *Bayesian model averaging* [Hoeting et al., 1999; Clyde, 1999].

Model Selection

Given prior $w_{\mathbb{M}}$ over the models, one way to explain the data using Bayesian inference is by selecting the model with maximum posterior probability:

$$\begin{aligned}\arg\max_{\mathcal{M}\in\mathbb{M}} w_{\mathbb{M}}(\mathcal{M}|x^n) &= \arg\max_{\mathcal{M}\in\mathbb{M}} \frac{P_{\mathcal{M}}(x^n)w_{\mathbb{M}}(\mathcal{M})}{\sum_{\mathcal{M}'\in\mathbb{M}} P_{\mathcal{M}'}(x^n)w_{\mathbb{M}}(\mathcal{M}')} \\ &= \arg\max_{\mathcal{M}\in\mathbb{M}} P_{\mathcal{M}}(x^n)w_{\mathbb{M}}(\mathcal{M}). \end{aligned} \tag{1.13}$$

This is called *Bayes factors* model selection [Kass and Raftery, 1995; Berger and Pericchi, 2001]. There are other methods as well.

1.4.2 Minimum Description Length

MDL views each $P_\theta \in \mathcal{M}$ as a code that assigns codelength $-\log P_\theta^n(x^n)$ to data x^n. As MDL is only interested in codelengths, it does not matter which actual code realises these codelengths as long as one exists. From an MDL point of view, each code describes the data. It can be more or less efficient as measured by its codelength for the data, but there is no notion of whether it is true in any sense. In the MDL view a model $\mathcal{M}$ is interpreted as a set of codes for the data that represents the conjecture that one of them will efficiently describe the data. It therefore constructs a universal model $P_{\mathcal{M}}^n$ for $\mathcal{M}$ for each sample size n. We will let $P_{\mathcal{M}} := P_{\mathcal{M}}^1, P_{\mathcal{M}}^2, \cdots$ denote the resulting sequence of universal models. Note that $P_{\mathcal{M}}$ is only a probabilistic source if the universal models in $P_{\mathcal{M}}$ are compatible (see page 4), which is not true for all possible choices of universal models. We will first discuss choices for $P_{\mathcal{M}}$ and then describe the MDL approach to combining the models in prediction and model selection.

Optimal Universal Model

In MDL the efficiency of a universal model on a given data sequence x^n relative to some set of codes is measured using its *regret*, which is the difference in codelength on x^n with the best code in the set. Thus the regret on x^n of a universal model $P_{\mathcal{M}}$ relative to model $\mathcal{M}$ is defined as:

$$R(P_{\mathcal{M}}, x^n) := -\log P_{\mathcal{M}}(x^n) - \min_{P_\theta \in \mathcal{M}} \{-\log P_\theta(x^n)\}.$$

$P_{\mathcal{M}}$ may have small or even negative regret on some data sequences x^n, but high regret on other sequences. To avoid assumptions about which data will actually be observed, $P_{\mathcal{M}}$ should have small regret on all possible data sequences. Its overall performance is therefore measured by its *worst-case* regret

$$R_{\max}(P_{\mathcal{M}}) := \max_{x^n \in \mathcal{X}^n} R(P_{\mathcal{M}}, x^n).$$

Under regularity conditions on $\mathcal{M}$, it can be shown that for each sample size n there exists a unique universal model P^n_{nml} that minimises $R_{\max}$ [Grünwald et al., 2005]. P^n_{nml} is given by

$$P^n_{\mathrm{nml}}(x^n) := \frac{\max_{P_\theta \in \mathcal{M}} P(x^n)}{\sum_{y^n \in \mathcal{X}^n} \max_{P_\theta \in \mathcal{M}} P_\theta(y^n)}.$$

It is called the normalised maximum likelihood (NML) distribution and achieves the same regret,

$$R(P^n_{\mathrm{nml}}, x^n) = \log \sum_{y^n \in \mathcal{X}^n} \max_{P_\theta \in \mathcal{M}} P_\theta(y^n),$$

on all sequences x^n. Unfortunately, the distributions P^1_{nml}, P^2_{nml}, $\cdots$ are not compatible, although for large sample sizes they are *almost* compatible. Therefore P_{nml} is not a probabilistic source and does not define a sequence of conditional distributions that can be used for the prediction of the next unobserved outcome. In addition it often cannot be computed efficiently. Under regularity conditions on $\mathcal{M}$ that will always be satisfied in this thesis, however, P_{nml} is approximated to order $o(1)$ by the Bayesian universal models $P_{\mathcal{M}}$ in (1.9) with Jeffreys' prior [Grünwald et al., 2005]. We will therefore always use Bayesian universal models based on Jeffreys' prior to approximate P_{nml}.

Prediction

MDL prediction amounts to the construction of a single conditional code C_{MDL} for the next outcome x_{n+1} given data x^n. This code is constructed by combining the codes for the models. It is based on the sequence of Bayesian universal models $P_{\mathrm{MDL}} = P^1_{\mathrm{MDL}}, P^2_{\mathrm{MDL}}, \cdots$ for the set $\{P_{\mathcal{M}} : \mathcal{M} \in \mathbb{M}\}$ that are defined by

$$P^n_{\mathrm{MDL}}(x^n) := \sum_{\mathcal{M}\in\mathbb{M}} P_{\mathcal{M}}(x^n) w_{\mathbb{M}}(\mathcal{M}) \tag{1.14}$$

for prior $w_{\mathbb{M}}(\mathcal{M})$ over the models. As we have made sure that each $P_{\mathcal{M}}$ is a probabilistic source, it follows by the same reasoning as in (1.10) that the sequence P_{MDL} is also a probabilistic source. C_{MDL} is now chosen such that it assigns codelength

$$L_{\mathrm{MDL}}(x_{n+1}|x^n) := -\log P_{\mathrm{MDL}}(x_{n+1}|x^n) \tag{1.15}$$

to the next outcome. By construction the distribution that corresponds to $C_{\mathrm{MDL}}(x_{n+1}|x^n)$ is $P_{\mathrm{MDL}}(x_{n+1}|x^n)$. MDL therefore predicts x_{n+1} according to $P_{\mathrm{MDL}}(x_{n+1}|x^n)$, which by a similar derivation to (1.12) can be computed as

$$P_{\mathrm{MDL}}(x_{n+1}|x^n) = \sum_{\mathcal{M}\in\mathbb{M}} P_{\mathcal{M}}(x_{n+1}|x^n) \frac{P_{\mathcal{M}}(x^n) w_{\mathbb{M}}(\mathcal{M})}{\sum_{\mathcal{M}'\in\mathbb{M}} P_{\mathcal{M}'}(x^n) w_{\mathbb{M}}(\mathcal{M}')}. \tag{1.16}$$

Model Selection

In model selection MDL also requires the construction of a single code for the data that combines the separate codes for the models. This time, however, it must explicitly identify a single model as the best explanation for the data. To satisfy this constraint MDL uses a two-part code that first identifies a single model and then codes the data with the help of that model:

$$\arg\min_{\mathcal{M}\in\mathbb{M}} \; -\log w_{\mathbb{M}}(\mathcal{M}) - \log P_{\mathcal{M}}(x^n). \tag{1.17}$$

It outputs the model that is thus selected.

1.4.3 Relating MDL and Bayes

Comparison of (1.12) and (1.16) reveals a close correspondence between MDL and Bayesian prediction: if the same priors w are used for each model and the same prior is used for $w_{\mathbb{M}}(\mathcal{M})$, then MDL and Bayesian prediction are equivalent. Furthermore, by monotonicity of the logarithm (1.13) can be rewritten as

$$\begin{aligned}\arg\max_{\mathcal{M}\in\mathbb{M}} P_{\mathcal{M}}(x^n)w_{\mathbb{M}}(\mathcal{M}) &= \arg\max_{\mathcal{M}\in\mathbb{M}} \log P_{\mathcal{M}}(x^n) + \log w_{\mathbb{M}}(\mathcal{M}) \\ &= \arg\min_{\mathcal{M}\in\mathbb{M}} -\log w_{\mathbb{M}}(\mathcal{M}) - \log P_{\mathcal{M}}(x^n),\end{aligned}$$

which under the same conditions equals Equation (1.17). In this case MDL and Bayesian model selection are therefore equivalent as well.[2]

In this thesis we will always use Jeffreys' prior for w. We will use a uniform prior $w_{\mathbb{M}}$ over the models [e.g. rejoinder in Hoeting et al., 1999]. In this case (1.12) and (1.16) reduce to

$$P_{\mathrm{MDL}}(x_{n+1}|x^n) = \sum_{\mathcal{M}\in\mathbb{M}} P_{\mathcal{M}}(x_{n+1}|x^n)\frac{P_{\mathcal{M}}(x^n)}{\sum_{\mathcal{M}'\in\mathbb{M}} P_{\mathcal{M}'}(x^n)},$$

and Equations (1.13) and (1.17) become equivalent to

$$\arg\min_{\mathcal{M}\in\mathbb{M}} \; -\log P_{\mathcal{M}}(x^n).$$

1.4.4 Criteria

To compare the performance of MDL and Bayes on prediction to alternative procedures we will use predictive loss, which we have defined in Section 1.2.6. By the equivalence of accumulated predictive loss and codelength it is the most natural loss measure from an MDL perspective. As the observations are probabilistically generated, the predictive loss of a procedure on a single outcome is subject to large fluctuations. We will therefore examine the average predictive loss over (subsequences of) the data or evaluate predictive loss in probability or in expectation.

[2] The use of Bayesian universal models in MDL is only justified as an approximation of the optimal universal model P_{nml}, which depends on regularity conditions on the models. Therefore the equivalence between MDL and Bayes does *not* hold in general.

In addition to achieving small predictive loss, it is widely regarded as important that a procedure be *consistent* [e.g. Berger and Pericchi, 2001; Grünwald et al., 2005; Rissanen, 1986b]. We will not use consistency as a performance measure, but will refer to consistency of MDL in model selection to guide our intuition in some of our proofs. Roughly speaking, consistency requires that a model selection procedure selects a model containing the generating distribution, if such a model exists, with probability going to one as the sample size goes to infinity. The data may be sampled from a source that is not in any of the models, however. Or there may not even exist any source such that the data can meaningfully be said to be sampled from it. In these cases consistency does not impose any restrictions on model selection procedures.

1.5 Chapter Summary

In this chapter we have introduced the Minimum Description Length (MDL) principle, which equates learning with finding regularity in the data. As any regularity can be used to give a shorter description of the data, it concludes that the goal of learning should be to find a short description of the data. The MDL principle is formalised using codes. However, it is only interested in the length of codewords and never in the actual codes. By the Kraft inequality codelength functions are closely related to probability distributions. In fact, we have adopted defective distributions and idealised codelengths, which make the correspondence one-to-one.

MDL may be applied to the tasks of prediction and model selection. In the presence of multiple models, prediction is the task of predicting the next outcome x_{n+1} given data x^n and model selection requires to select the best explanation for the data. The resulting procedures are given by (1.16) and (1.17), respectively. We use Bayesian universal models based on Jeffreys' prior for the models, which are compatible approximations of the optimal NML models. As a consequence these procedures also have a Bayesian interpretation. In Bayesian statistics the log loss is a well-known measure to evaluate predictive procedures. In this context we call it predictive (log) loss. We have shown that the accumulated predictive loss over the data may also be viewed as codelength by the sequential decomposition of codelength. It is therefore a natural loss measure from an MDL perspective.

CHAPTER 2

The Momentum Problem

In the next section we introduce the momentum problem by a simple example that will be called the *Bernoulli example*. It presents a simple scenario involving two nested models that exhibits the so-called *momentum phenomenon*. In Section 2.2 we will introduce the Switch-Point procedure, which is designed to exploit the momentum phenomenon to improve predictive performance. Section 2.3 then confirms that the Switch-Point procedure does improve predictive performance slightly on the Bernoulli example. In the next chapter we will consider a more complicated example on which the Switch-Point procedure achieves a larger improvement. We conclude that the momentum phenomenon should be considered a momentum *problem*.

2.1 Bernoulli Example

Inspired by Dawid [1984], we consider two meteorologists issuing daily forecasts of the "probability of precipitation" for the next day in the same region. The next day they observe whether any precipitation actually falls and update their predictions accordingly. Each day we decide whether or not to bring an umbrella to work based on the forecasts by the two meteorologists and our knowledge of their accuracy on all preceding days.

Suppose that both meteorologists base their predictions on a model for the probability of precipitation. Then we are faced with a prediction problem

in the presence of multiple models. They may have come up with their models in different ways. One meteorologist, for instance, may have based his model on existing meteorologic records of precipitation in a nearby region; the other may have wanted to avoid any assumptions about the similarity of the weather between the two regions.

Let us consider a more formal version of this problem on binary data $x_1, \ldots, x_n$ that denote the occurrence of precipitation in a time-series. Suppose the models employed by the two meteorologists are the Single Bernoulli model, denoted by $\mathcal{M}_0$, and the Bernoulli model, denoted by $\mathcal{M}_1$, which were introduced in Section 1.3.1. To avoid confusion between the models we will frequently refer to $\mathcal{M}_1$ as the *Full* Bernoulli model.

We will use Jeffreys' prior to construct Bayesian universal models for $\mathcal{M}_1$ at each sample size. It is given by

$$w_{\text{Jeffreys}} = \frac{\theta^{-\frac{1}{2}}(1-\theta)^{-\frac{1}{2}}}{\pi}$$

[Grünwald, 2007]. $\mathcal{M}_0$ contains only a single probabilistic source $P_{0.6}$. By assumption Jeffreys' prior therefore assigns all prior probability to $P_{0.6}$. As a consequence the resulting universal model for $\mathcal{M}_0$ is equal to $P_{0.6}$ and achieves zero regret on all sequences. The universal models on any sequence of n observations x^n can now be computed by

$$P_{\mathcal{M}_0}(x^n) := 0.6^{n_1(x^n)} \cdot (1-0.6)^{n_0(x^n)},$$

and

$$P_{\mathcal{M}_1}(x^n) := \int_0^1 \frac{\theta^{-\frac{1}{2}} \cdot (1-\theta)^{-\frac{1}{2}}}{\pi} \cdot \theta^{n_1(x^n)} \cdot (1-\theta)^{n_0(x^n)} \, d\theta,$$

where $n_1(x^n)$ and $n_0(x^n)$ respectively denote the number of ones and the number of zeroes in x^n. We denote the corresponding codelengths by $L_0(x^n)$ and $L_1(x^n)$ respectively.

A possible objection to the Single Bernoulli model might be that it is unrealistic. Surely in practice the meteorologist considering it would rather model his assumptions by using the Full Bernoulli model with a strongly peaked prior around $\theta = 0.6$. This however would not significantly change the example; the observation of the momentum phenomenon that will be made below, would remain unchanged. For simplicity the Single Bernoulli model is therefore defined as it is.

Suppose we observe data x^n. Then by (1.6) the accumulated predictive loss of P_{MDL} on x^n is equal to $-\log P_{\mathrm{MDL}}(x^n)$. P_{MDL} is a Bayesian universal model for $P_{\mathcal{M}_0}$ and $P_{\mathcal{M}_1}$. As discussed in Section 1.2.7 it therefore behaves approximately like the model that assigns shortest codelength to x^n. To investigate the behaviour of P_{MDL} it will therefore be most informative to examine $L_0(x^n)$ and $L_1(x^n)$. In particular, we are interested in the difference $L_1(x^n) - L_0(x^n)$, which a Bayesian would interpret as the odds provided by the data for the Single Bernoulli model versus the Full Bernoulli model [Berger and Pericchi, 2001]. If $L_1(x^n) - L_0(x^n) > 0$ then the Single Bernoulli has best predicted the data; otherwise the Full Bernoulli model was the best predictor.

Let x^n be sampled sequentially from P_{θ^*} in the Full Bernoulli model. Then it will be informative to examine how $L_1(x^n)$ and $L_0(x^n)$ develop with n. We will therefore depict the difference $L_1(x^n) - L_0(x^n)$ for all sample sizes up to some reasonable n. For a first impression we will first investigate the difference in codelength between the models on typical individual sequences for various choices of P_{θ^*}. Then we will look at the difference in expectation to gain insight into its overall behaviour.

2.1.1 Results: Codelength on Individual Sequences

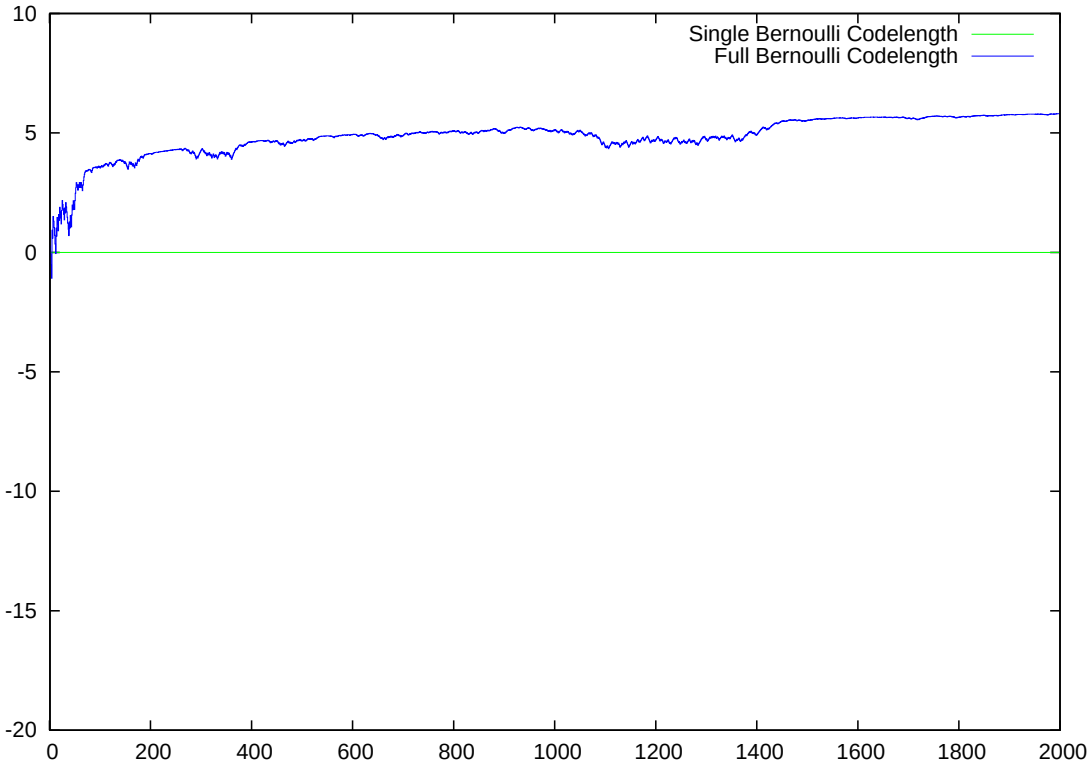
Figure 2.1 shows the difference in codelength between the two models on individual sequences that have been sampled from various generating sources. We have included many generating sources for ease of comparison with results in expectation in the next section. They have been sorted by their distance in the parameter space from the source in the Single Bernoulli model. All sequences have been generated in a single attempt. Although no strong conclusions can be justified based on the properties of probabilistically generated individual sequences, we do make the following preliminary observation: for some generating distributions the Single Bernoulli model is the best predictor on average for small sample sizes and the Full Bernoulli model is the best predictor on average for large sample sizes. In other words, the best-predicting model switches with increasing sample size. The effect appears to be larger for generating distributions that are more similar to the distribution in the Single Bernoulli model. For each sequence shown in the figure, ten alternative individual sequences were generated with the same parameter settings. In all cases our preliminary observation held. We will now make this observation more precise and argue that it suggests suboptimal behaviour of MDL and Bayes.

Consider Figures 2.2(b), 2.2(c), 2.2(d) and 2.2(e), which show sequences sampled from generating distributions in the Full Bernoulli model with parameter θ^* equal to 0.55, 0.65, 0.50 and 0.70 respectively. Due to the randomness in the sampling process for the sequences, the difference in codelength between the Full Bernoulli model and the Single Bernoulli model exhibits significant local fluctuations. We do, however, observe a pattern in these figures: in all cases the difference in codelength first increases to some maximum, and then continually decreases again. The pattern is absent in Figures 2.1(a), 2.2(f) and 2.2(g), which have been sampled from the distributions with θ^* equal to 0.60, 0.30 and 0.90 respectively. We call this pattern the *momentum phenomenon*. To be precise, let x^n be any data sequence and let $L_a(x^n)$ and $L_b(x^n)$ denote the codelength assigned to x^n by any two models $\mathcal{M}_a$ and $\mathcal{M}_b$ respectively. We then say that the momentum phenomenon occurs on x^n if there exist sample sizes n_1, n_2 with $n_1 < n_2$ such that $L_a(x^{n_1}) < L_b(x^{n_1}) - C$ and $L_a(x^{n_2}) > L_b(x^{n_2})$ for some constant $C > 0$.

The Single Bernoulli model must have achieved at least C bits shorter codelength than the Full Bernoulli model on the first n_1 observations, but the Full Bernoulli model must have achieved at least C bits shorter codelength on the part of the sequence between n_1 and n_2. That is, the Single Bernoulli was the best-predicting model on the first n_1 outcomes, but the Full Bernoulli model was the best-predicting model on the outcomes between n_1 and n_2. We call the maximum C for which x^n exhibits the momentum phenomenon the *size of the momentum phenomenon*. The size of the momentum phenomenon is equal to the maximum number of bits that can be gained by switching between models at exactly the right sample size. That is, it reflects how much we might hope to gain by exploiting the momentum phenomenon. Although we have seen that the momentum phenomenon does not always occur, we will prove in Section 3.1 that for each size C of the momentum phenomenon and for each $\epsilon > 0$ there exist generating distributions in the Full Bernoulli model such that the size of the momentum phenomenon is at least C with probability at least $1 - \epsilon$.

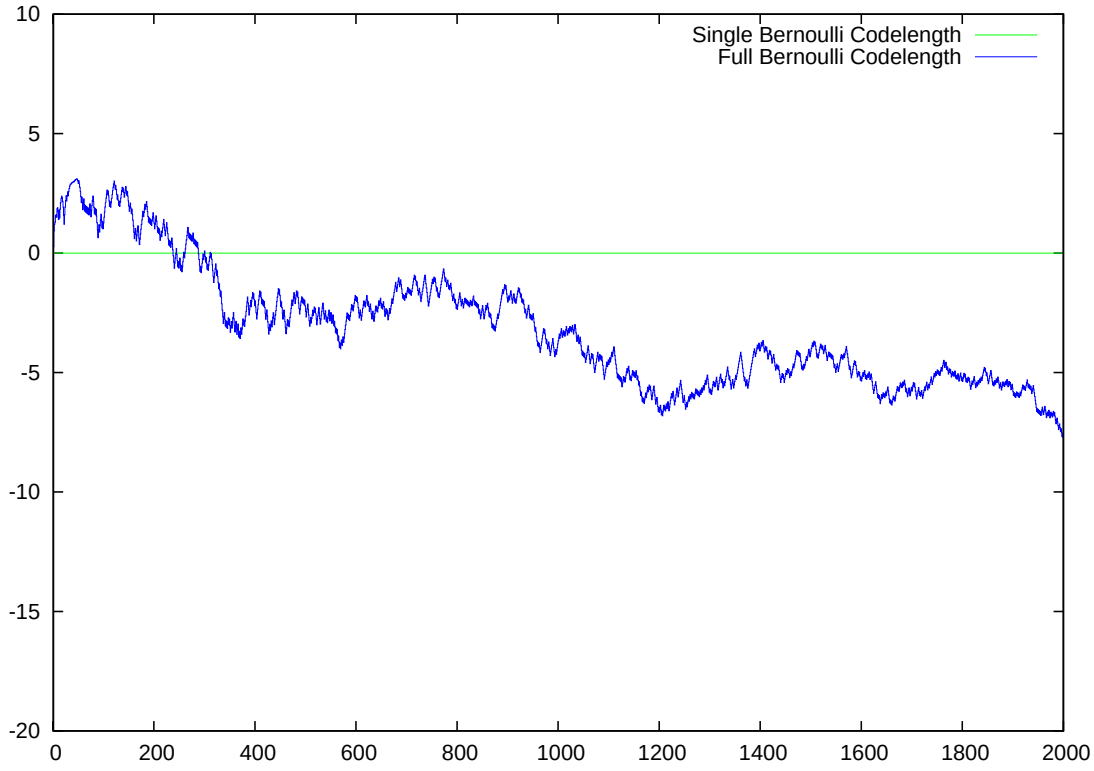
MDL and Bayes predict approximately (in the sense of page 15) like the model that best predicts the entire data x^n. However, if the best-predicting model changes over time from the Single Bernoulli model to the Full Bernoulli model, then it would be better to predict the first part of the data according to the Single Bernoulli model and the rest of the data according to the Full Bernoulli model. When this happens, the Full Bernoulli model has to make up for its initial poor performance before it starts outperforming the Single Bernoulli model on all data. Meanwhile, MDL and Bayesian predictions still

resemble the predictions of the *old* best-predicting model much more closely than they resemble the predictions of the *currently* best-predicting model. This will be illustrated later. We might say that MDL and Bayes have to overcome their momentum towards the Single Bernoulli model before they start predicting according to the Full Bernoulli model. This motivates the naming of the momentum phenomenon.

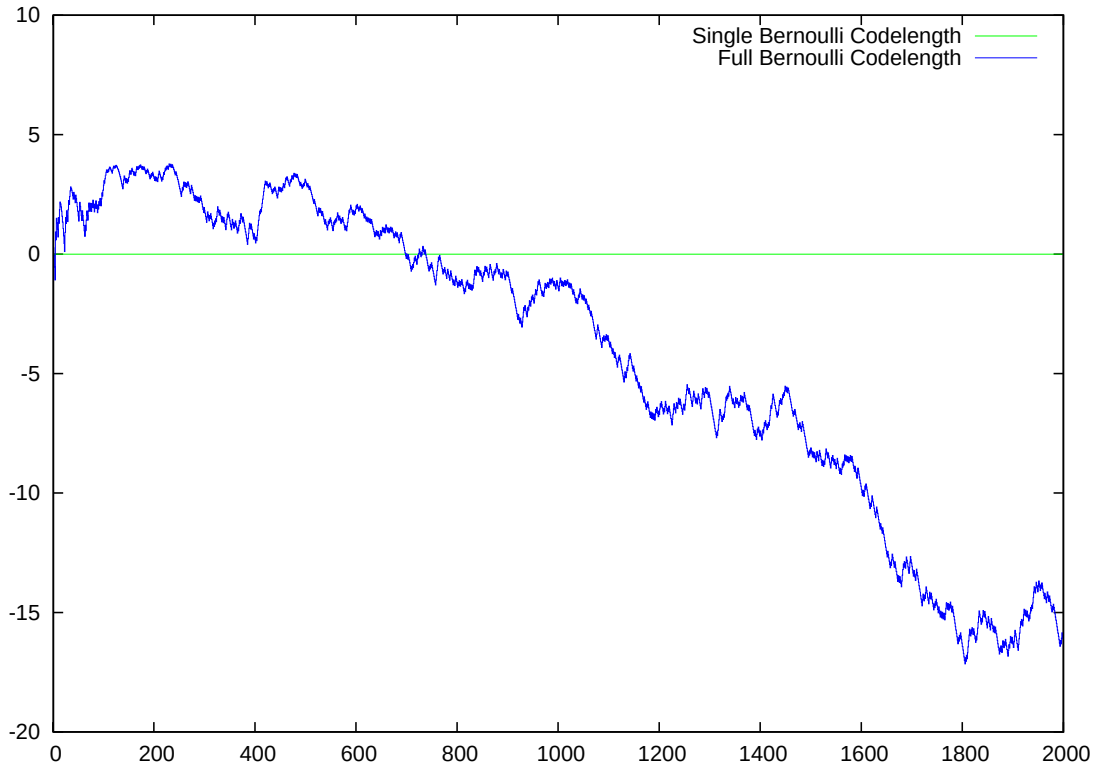


(a) $\theta^* = 0.6$

Figure 2.1: $L_1(x^n) - L_0(x^n)$ on sequences x^n sampled from the Bernoulli distribution with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 2000$.

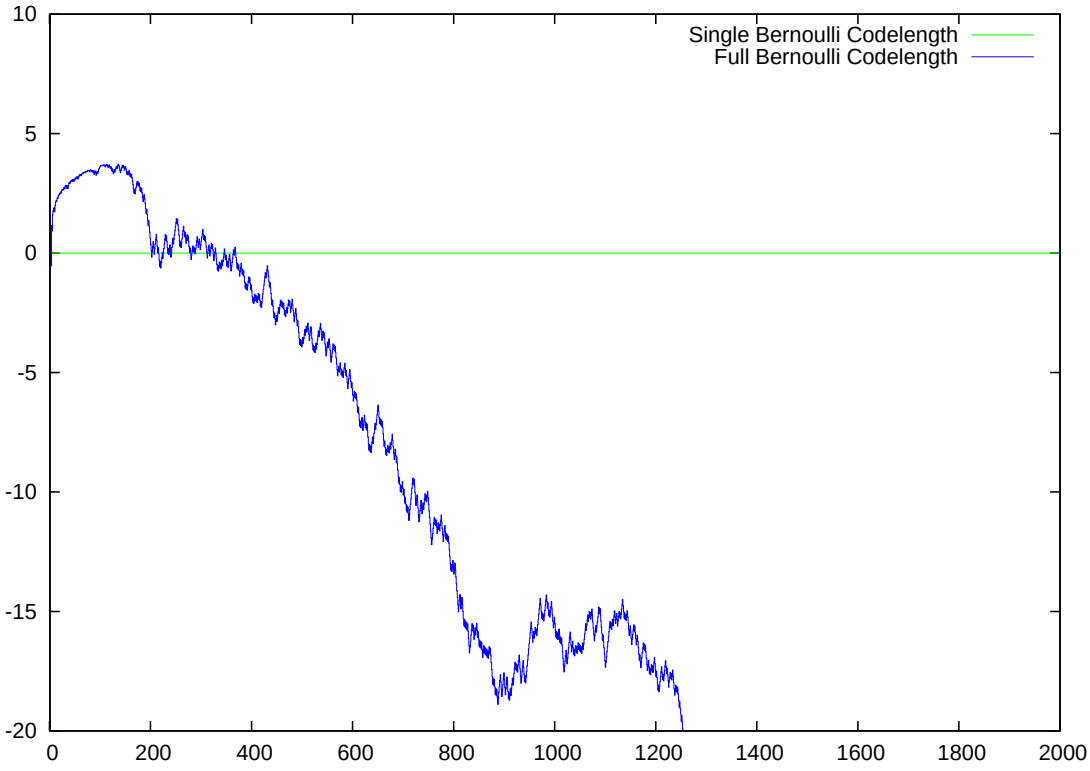


(b) $\theta^* = 0.55$

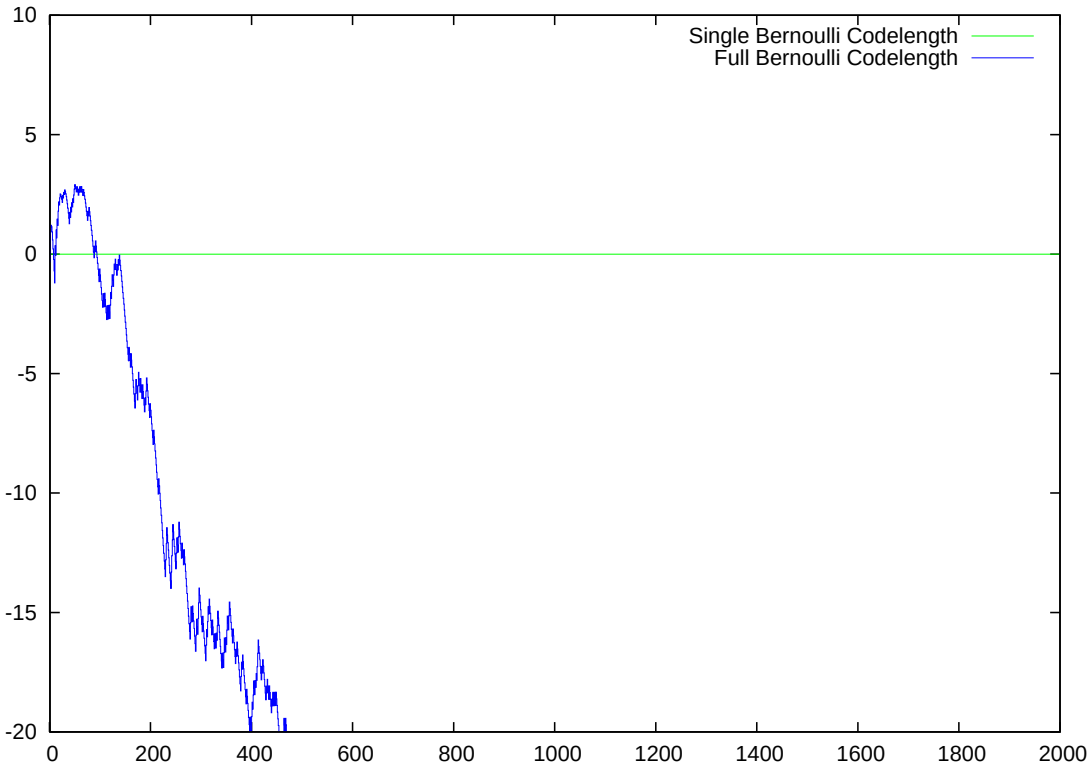


(c) $\theta^* = 0.65$

Figure 2.1 (cont.): $L_1(x^n) - L_0(x^n)$ on sequences x^n sampled from the Bernoulli distribution with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 2000$.

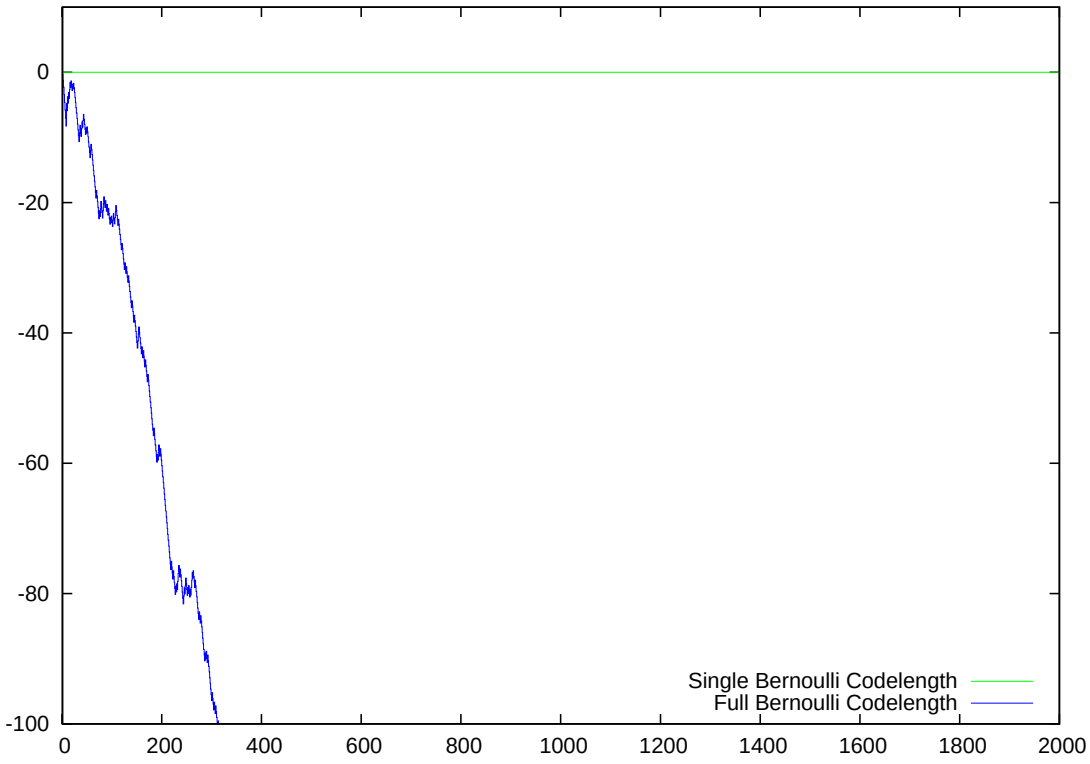


(d) $\theta^* = 0.5$

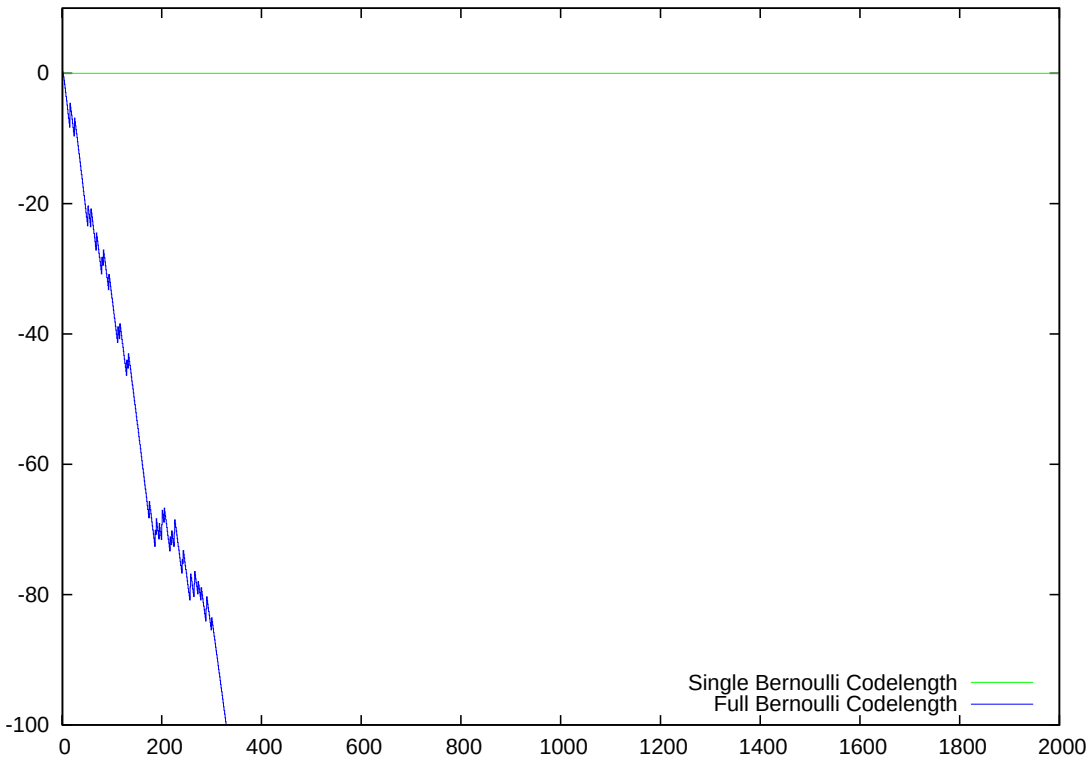


(e) $\theta^* = 0.7$

Figure 2.1 (cont.): $L_1(x^n) - L_0(x^n)$ on sequences x^n sampled from the Bernoulli distribution with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 2000$.



(f) $\theta^* = 0.3$



(g) $\theta^* = 0.9$

Figure 2.1 (cont.): $L_1(x^n) - L_0(x^n)$ on sequences x^n sampled from the Bernoulli distribution with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 2000$. N.B. The scale of the vertical axis in the figures on this page differs from the scale in the previous figures!

2.1.2 Results: Expected Codelength

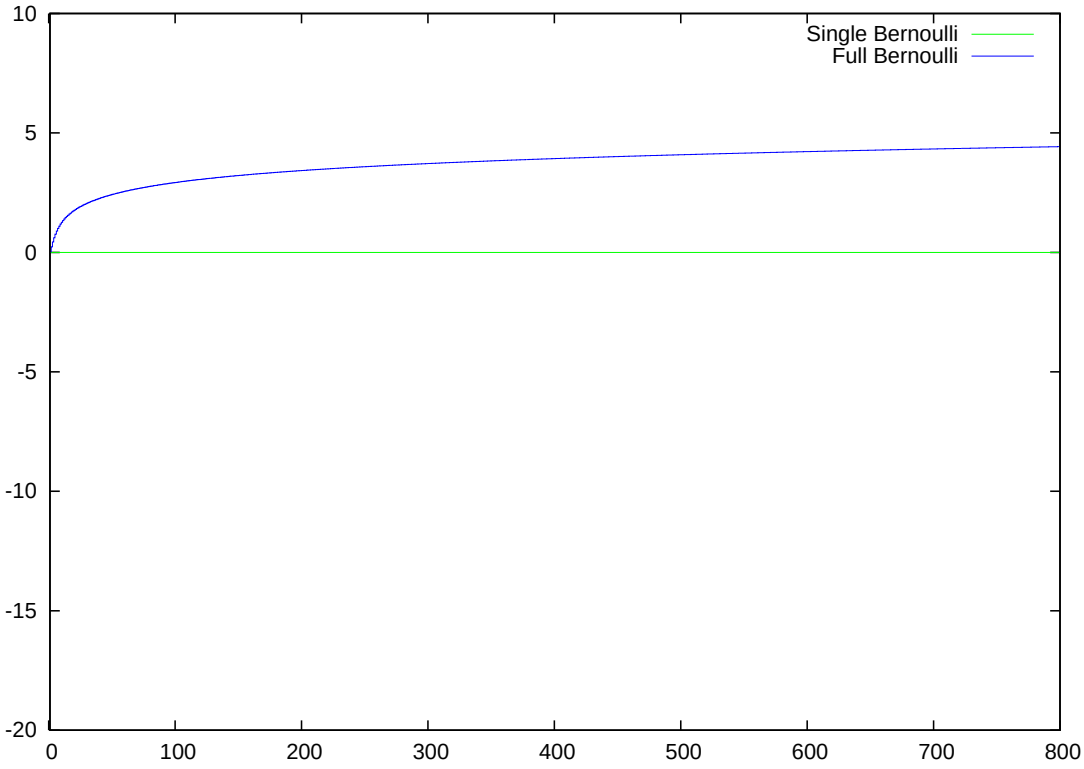
Figure 2.2 shows the expected difference in codelength between the two models under the same generating distributions as were used to sample the individual sequences. We note that by linearity of expectation the expected difference in codelength between two models is equal to the difference in expected codelength. Due to computational limitations the range of sample sizes shown has been reduced compared to the range for individual sequences.

It can be seen that which model predicts best in expectation changes in the same way as for the individual sequences from the previous section. The expected codelength of the models, however, is not random and therefore does not exhibit any local fluctuations. Figures 2.3(b), 2.3(c), 2.3(d) and 2.3(e) therefore suggest that the momentum phenomenon occurs for many sequences generated by the distributions in the Full Bernoulli model with θ^* equal to 0.55, 0.65, 0.50 and 0.70 respectively. In fact, it will be proved in Section 3.1 that the momentum phenomenon gets arbitrarily large under some generating distributions in the Full Bernoulli model with arbitrarily high probability for (sufficiently large) sample sizes.

We observe that the momentum phenomenon occurs whenever the generating distribution is sufficiently similar to the distribution in the Single Bernoulli model, but not if the two are actually equal. This can be explained by different rates at which the parameters for the models are learned. Model $\mathcal{M}_0$ contains no parameters and therefore the associated sequential prediction algorithm, which predicts according to $P_{\mathcal{M}_0}(x_{n+1}|x^n)$, does not need to learn their values. Model $\mathcal{M}_1$, by contrast, does contain a parameter and $P_{\mathcal{M}_1}(x_{n+1}|x^n)$ requires a significant number of observations before its (implicit) parameter estimate becomes reasonably accurate. For small sample sizes the overhead for learning optimal parameter values contributes significantly to the codelengths of the models. For large sample sizes, however, the codelengths of the models are dominated by the codelength assigned to the data by the best source in the models as measured by Kullback-Leibler divergence from the generating source [Berger and Pericchi, 2001]. The momentum phenomenon arises when the model containing the best source among all models converges slower than another model and the difference in overhead for parameter learning between the two models exceeds the difference in codelength between the best sources in the models for small sample sizes.

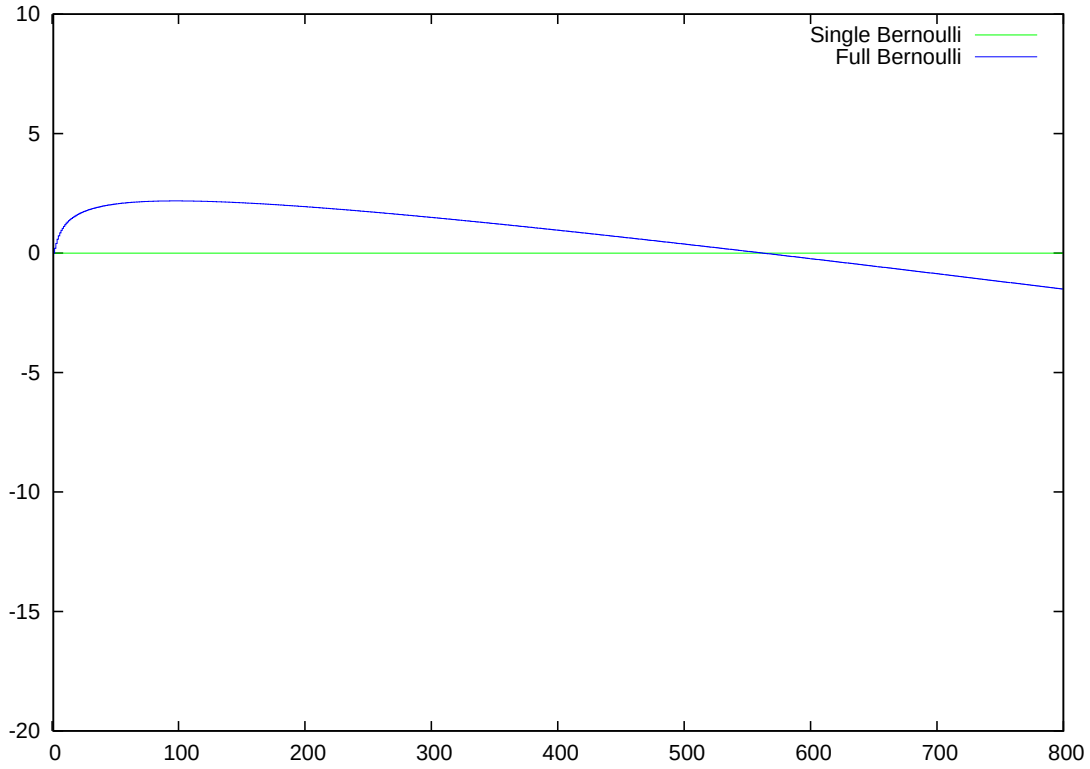
It is not immediately clear that the momentum phenomenon poses a problem: we cannot expect any method to approximate the best-predicting model

under all circumstances. However, in the next section we will introduce the Switch-Point procedure, which attempts to exploit the momentum phenomenon by explicitly modelling at which sample size we should switch from one model to the other. We will demonstrate small gains by the Switch-Point procedure on the current example in Section 2.3 and significant gains on a more complicated example in Section 3.3. If the momentum phenomenon is absent, then the codelength of the Switch-Point procedure is within a constant —less than 0.6 bits in the worst-case — of the MDL codelength, which makes it nearly as efficient as MDL and Bayes. Thus the Switch-Point procedure always predicts nearly as well as MDL and Bayes, but sometimes predicts significantly better. The improved predictive performance of the Switch-Point procedure demonstrates that both MDL and Bayes deal with the momentum phenomenon suboptimally. We call this fact the *momentum problem*.

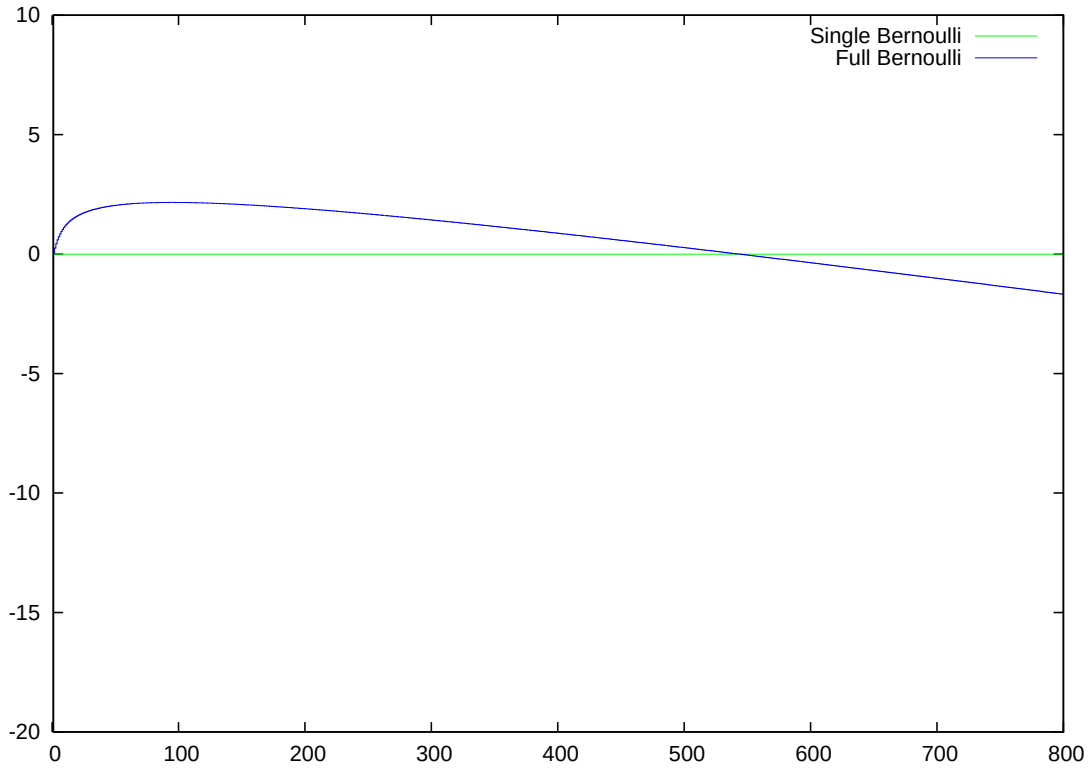


(a) $\theta^* = 0.6$

Figure 2.2: $E_{P_{\theta^*}}[L_1(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ under Bernoulli distribution $P_{\theta^*} \in \mathcal{M}_1$ with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 800$.

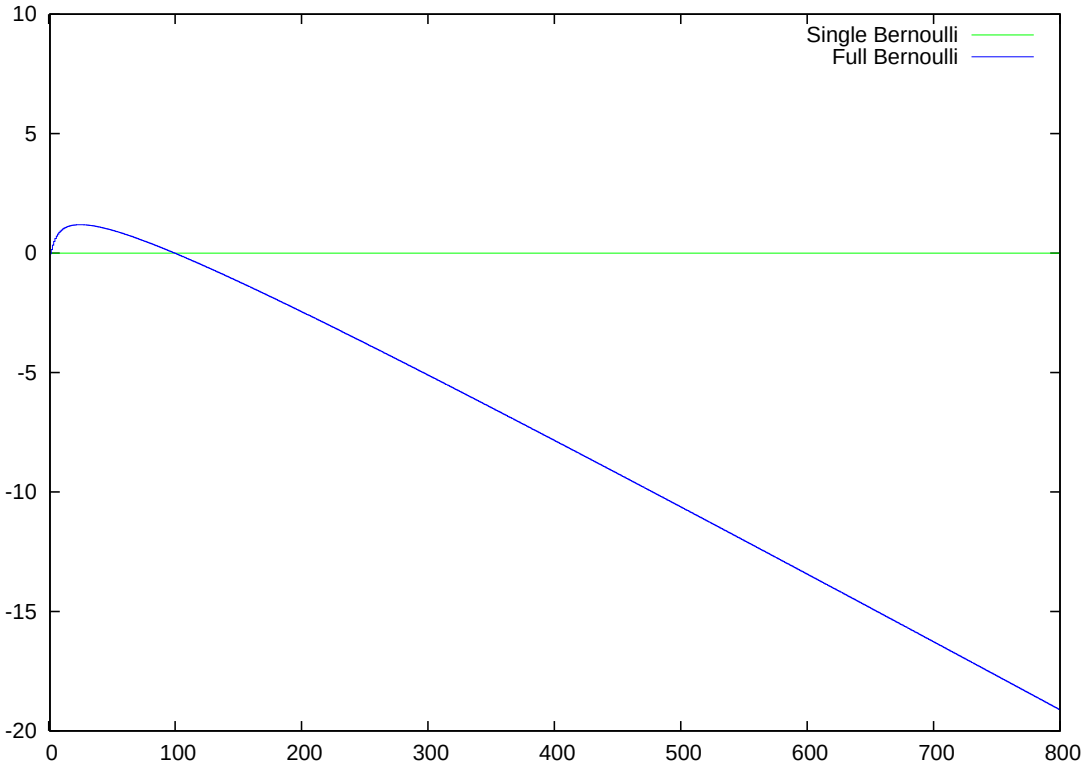


(b) $\theta^* = 0.55$

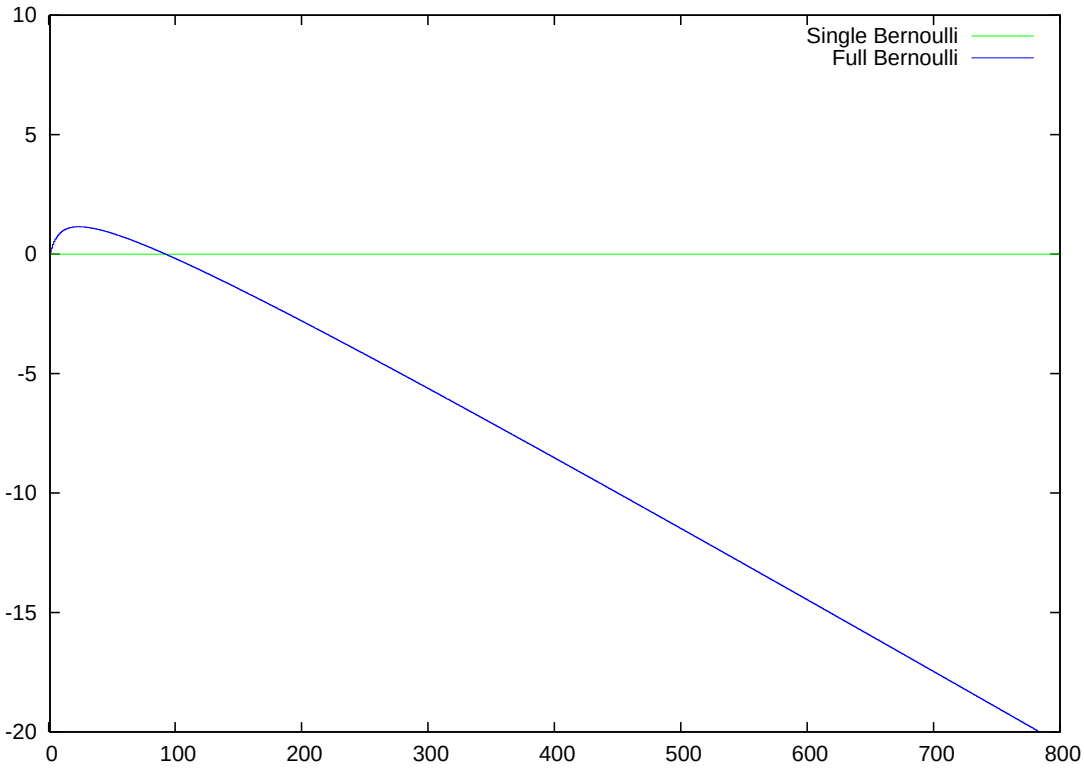


(c) $\theta^* = 0.65$

Figure 2.2 (cont.): $E_{P_{\theta^*}}[L_1(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ under Bernoulli distribution $P_{\theta^*} \in \mathcal{M}_1$ with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 800$.

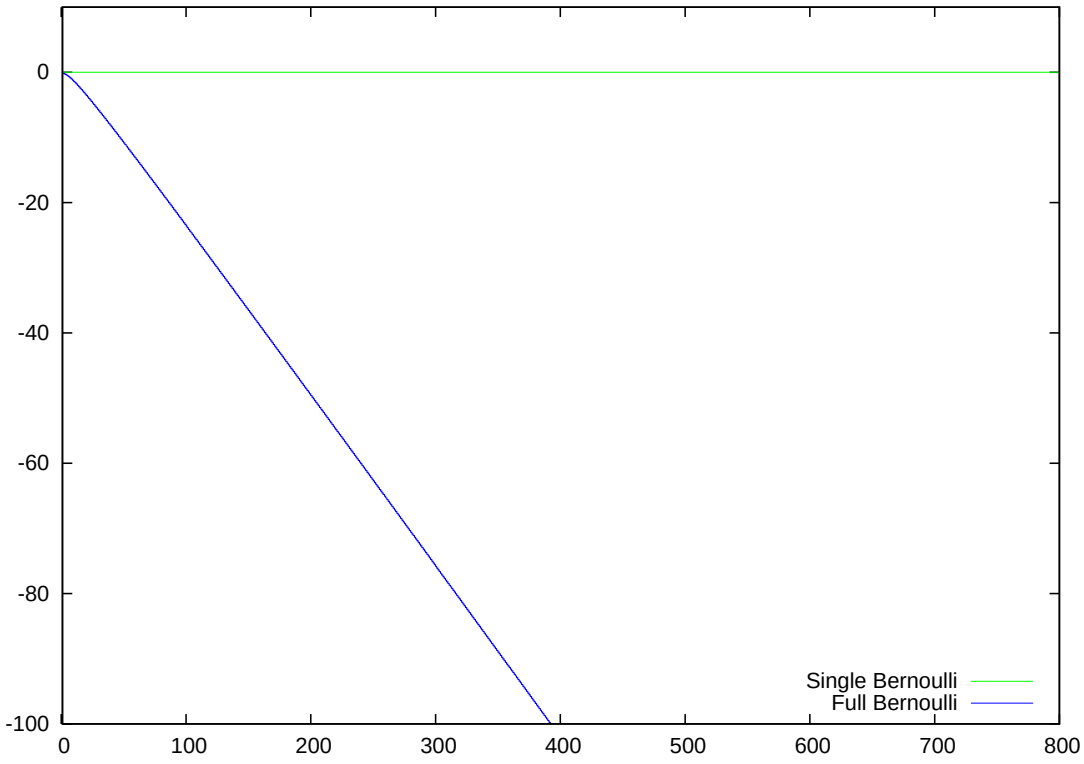


(d) $\theta^* = 0.5$

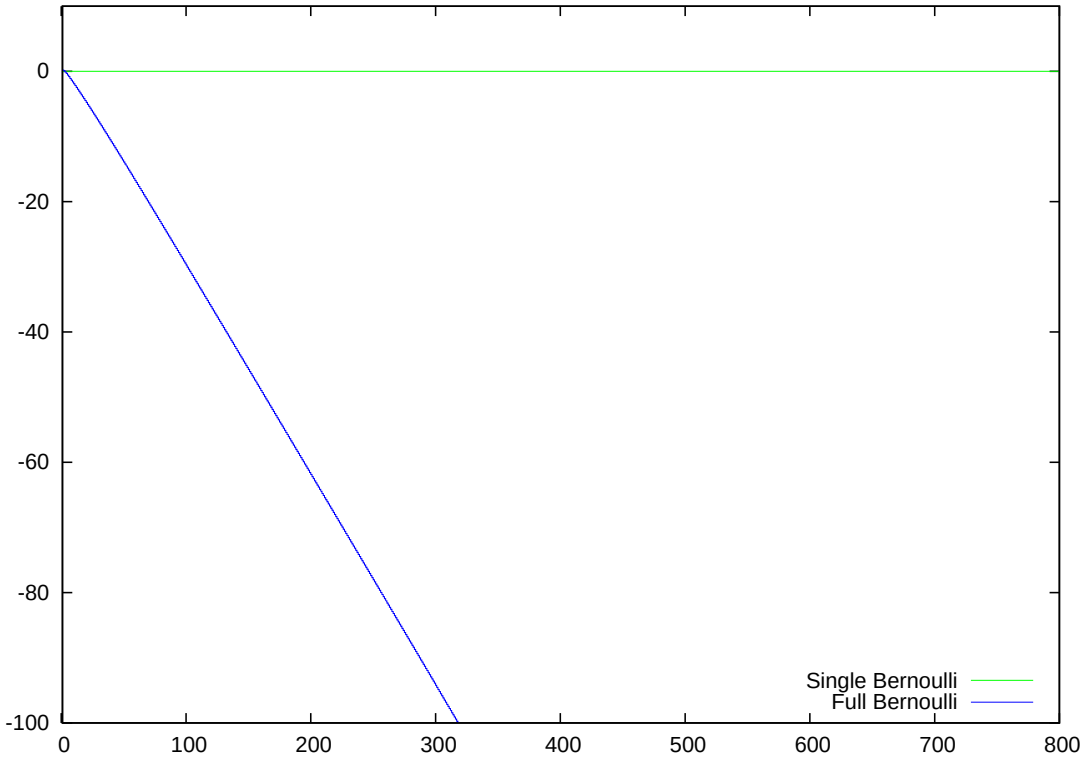


(e) $\theta^* = 0.7$

Figure 2.2 (cont.): $E_{P_{\theta^*}}[L_1(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ under Bernoulli distribution $P_{\theta^*} \in \mathcal{M}_1$ with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 800$.



(f) $\theta^* = 0.3$



(g) $\theta^* = 0.9$

Figure 2.2 (cont.): $E_{P_{\theta^*}}[L_1(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ under Bernoulli distribution $P_{\theta^*} \in \mathcal{M}_1$ with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 800$. N.B. The scale of the vertical axis in the figures on this page differs from the scale in the previous figures!

2.2 Switch-Point Procedure

This section describes the *Switch-Point* procedure, which is designed to deal explicitly with the momentum phenomenon in prediction in the presence of two models. It applies when the occurrence of the momentum phenomenon cannot be ruled out as, for instance, in the Bernoulli example. Let the two models be labelled $\mathcal{M}_a$ and $\mathcal{M}_b$ with Bayesian universal models $P_{\mathcal{M}_a}$ and $P_{\mathcal{M}_b}$ and let the corresponding codelengths for any data sequence x^n be denoted by $L_a(x^n) := -\log P_{\mathcal{M}_a}(x^n)$ and $L_b(x^n) := -\log P_{\mathcal{M}_b}(x^n)$ respectively. Furthermore, suppose that we are to encode a sequence of n observations in a time-series. We would like to assign short codelength to the sequence if all observations in the sequence are predicted well by a single model; or if the model that tends to best predict the next observation changes from $\mathcal{M}_a$ to $\mathcal{M}_b$ somewhere in the sequence. The first case is handled well by regular MDL and Bayes. The second deals with the momentum phenomenon.

The two stated goals are accomplished by adding an extra model $\mathcal{M}_{\mathrm{s}}$ — subscript s for Switch-Point model — that assigns short codelength to the sequence if the best predicting model on the sequence changes from model $\mathcal{M}_a$ to model $\mathcal{M}_b$ and applying MDL or Bayes to the new problem of predicting with models $\mathcal{M}_a$, $\mathcal{M}_b$ and $\mathcal{M}_{\mathrm{s}}$. This procedure is called the Switch-Point procedure after the Switch-Point model $\mathcal{M}_{\mathrm{s}}$ that it adds. The code for the Switch-Point model is called the *Switch-Point code*. Its codelength, called the *Switch-Point codelength*, on any sequence x^n is denoted by $L_{\mathrm{s}}(x^n) = -\log P_{\mathcal{M}_{\mathrm{s}}}(x^n)$. In general the codelength assigned to any sequence of observations by MDL or Bayes is guaranteed to be approximately equal to the codelength of the best model. It will be shown in Section 4.2 that the addition of the extra model $\mathcal{M}_{\mathrm{s}}$ can in the worst case increase the MDL or Bayes codelength by a constant c, which depends only on the number of models and not on the data. Under a uniform prior $w_{\mathbb{M}}$ over the models, $c \leq -\log \frac{2}{3} \approx 0.58$ if the number of models is increased from two to three. This worst-case is achieved only if $P_{\mathcal{M}_{\mathrm{s}}}(x^n) = 0$. The addition of the extra model can therefore never significantly increase the total codelength assigned to a sequence of observations. On the other hand, if $L_{\mathrm{s}}(x^n)$ is shorter than both $L_a(x^n)$ and $L_b(x^n)$, then the addition of the Switch-Point model reduces the total codelength for x^n. This follows because in this case $P_{\mathcal{M}_{\mathrm{s}}}(x^n)$

is larger than $P_{\mathcal{M}_a}(x^n)$ and $P_{\mathcal{M}_b}(x^n)$, which implies that

$$-\log \sum_{\mathcal{M}\in\mathbb{M}\cup\{\mathcal{M}_\mathrm{s}\}} \frac{1}{3}\cdot P_{\mathcal{M}}(x^n) = -\log\left(\frac{2}{3}\cdot P_{\mathrm{MDL}}(x^n) + \frac{1}{3}\cdot P_{\mathcal{M}_\mathrm{s}}(x^n)\right)$$
$$< -\log P_{\mathrm{MDL}}(x^n).$$

It follows that the Switch-Point procedure satisfies the two goals stated above if a suitable model $\mathcal{M}_\mathrm{s}$ can be constructed.

How should model $\mathcal{M}_\mathrm{s}$ be constructed? Recall that its codes should assign short codelength to the data if the best predicting model changes from $\mathcal{M}_a$ to $\mathcal{M}_b$. Each code should code the start of the sequence using $P_{\mathcal{M}_a}$ and after some unknown — possibly zero — number of observations s it should *switch* to code the remainder of the sequence with $P_{\mathcal{M}_b}$. $\mathcal{M}_\mathrm{s}$ is constructed as the set of such codes for all possible switch-points s and a Bayesian universal model $P_{\mathcal{M}_\mathrm{s}}$ is constructed using some reasonably flat prior w_s over the switch-points. Thus, the resulting codelength for $P_{\mathcal{M}_\mathrm{s}}$ is given by

$$L_\mathrm{s}(x^n) := -\log \sum_{s=0}^{\infty} w_s(s)\cdot P_{\mathcal{M}_a}(x^{\min(s,n)})\cdot P_{\mathcal{M}_b}(x^n|x^{\min(s,n)}), \tag{2.1}$$

where $P_{\mathcal{M}_a}(x^{\min(s,n)})\cdot P_{\mathcal{M}_b}(x^n|x^{\min(s,n)})$ is the distribution corresponding to a code that switches from model $\mathcal{M}_a$ to model $\mathcal{M}_b$ at switch-point s. When s exceeds the sample size, n, all outcomes are coded using $P_{\mathcal{M}_a}$. We therefore take the minimum of s and n. By virtue of its construction as a Bayesian universal model, $L_\mathrm{s}(x^n)$ will never exceed the codelength for the optimal switch-point $\hat{s}$ by more than $-\log w_s(\hat{s})$ bits. Throughout this thesis we will use

$$w_s(s) := \frac{s^{-1.1}}{10.5844}$$

for the prior over the switch-points.

Alternatively, one might consider constructing $\mathcal{M}_\mathrm{s}$ using different mixtures of $P_{\mathcal{M}_a}$ and $P_{\mathcal{M}_b}$. We will show in Section 4.2 that any set of fixed-ratio mixtures is not a suitable choice for $\mathcal{M}_\mathrm{s}$.

2.3 Switch-Point on the Bernoulli Example

The Switch-Point procedure will now be applied to the Bernoulli example. It will be shown that the inclusion of the Switch-Point code can shorten

codelengths by mitigating the effects of the momentum phenomenon. The number of bits that can be won by the Switch-Point code depends on the size of the momentum phenomenon. In the Bernoulli example the momentum phenomenon is small. If the Switch-Point code is able to achieve shorter codelength at all in this example, then this may therefore be considered significant and a sign of its efficiency in exploiting the momentum phenomenon. Section 3.3 will present a (more complicated) example in which the reduction in codelength is much larger.

We have observed in Section 2.1 that the best predicting model changed from the Single Bernoulli model to the Full Bernoulli model if the data were sampled from any of a number of generating distributions in the Full Bernoulli model. We therefore let model $\mathcal{M}_a$ correspond to the Single Bernoulli model and model $\mathcal{M}_b$ to the Full Bernoulli model. That is, the Switch-Point code switches from the Single Bernoulli model to the Full Bernoulli model. The following two sections will compare the Switch-Point codelength to the codelengths of the two original models in the Bernoulli example. We will revisit the individual sequences from Section 2.1 as well as compare codelengths in expectation.

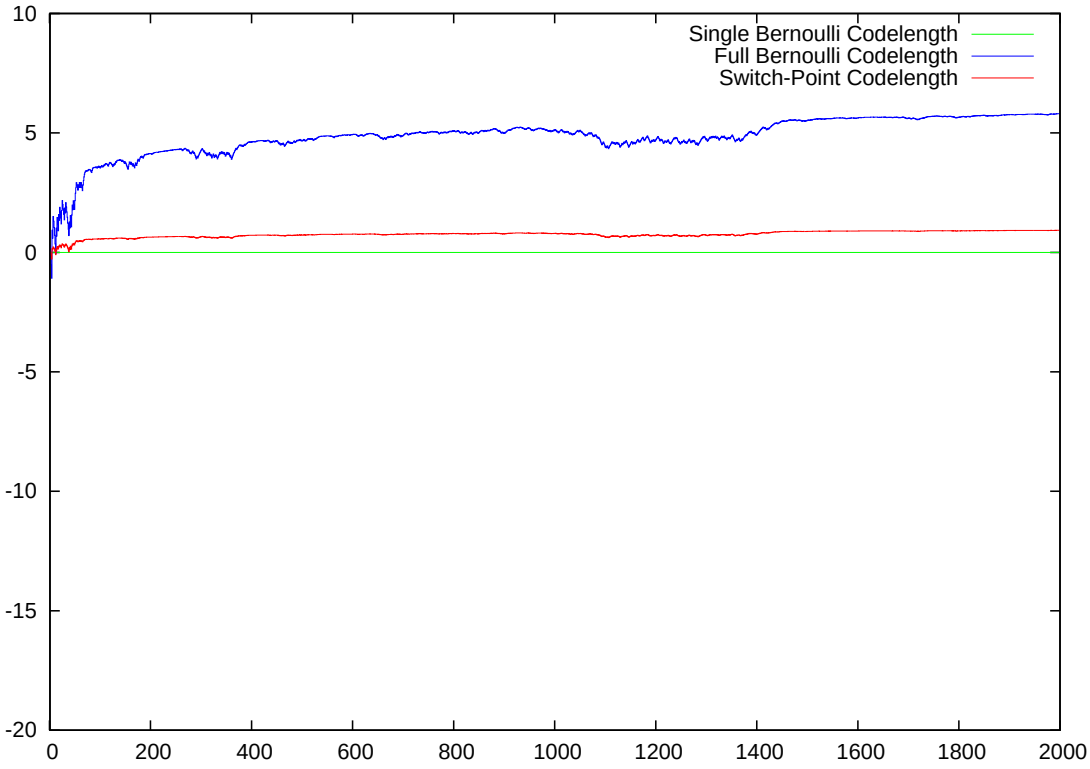
2.3.1 Results: Switch-Point on Individual Sequences

For a first impression of the behaviour of the Switch-Point code on the Bernoulli example, we compare the Switch-Point codelength to the codelengths of the two Bernoulli models on the individual sequences from Figure 2.1, which is repeated in Figure 2.3 with the Switch-Point codelength added. For ease of comparison the codelength according to the Single Bernoulli model has again been subtracted.

The momentum phenomenon occurred for the sequences sampled from generating distributions in the Full Bernoulli model with parameter θ^* equal to 0.55, 0.65, 0.50 and 0.70. The corresponding figures are Figures 2.4(b), 2.4(c), 2.4(d) and 2.4(e) respectively. For the first three of these sequences we now observe that the Switch-Point code achieves slightly shorter codelength than the two Bernoulli codes for all shown sample sizes greater than a certain sample-size. In all cases it achieves shorter codelength than the Single Bernoulli model already at smaller sample sizes than the Full Bernoulli model. For the sequence sampled from generating distribution with θ^* equal

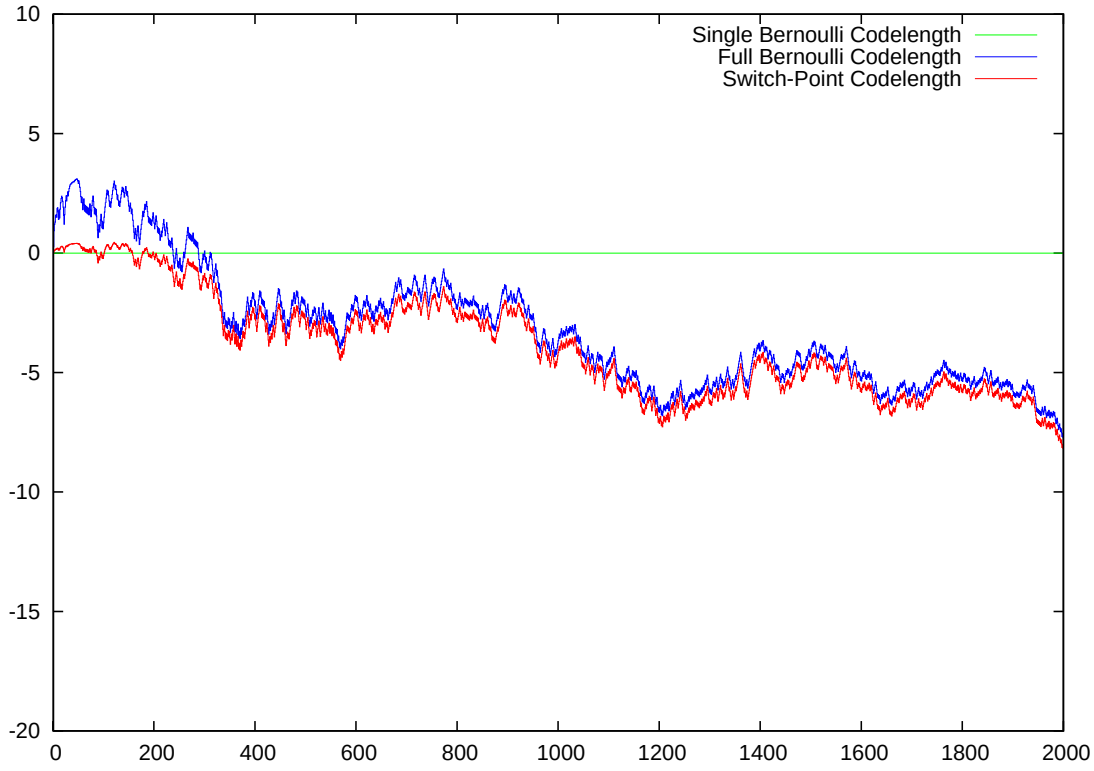
to 0.70 the Switch-Point code achieves slightly worse codelength — approximately 0.12 bits for all sample sizes greater than 220 — than the Full Bernoulli model. It achieves much better codelength — more than 10 bits for the same sample sizes — than the Single Bernoulli model. Explicit computation of $-\log P_{\mathrm{MDL}}$ and the codelength of the Switch-Point procedure shows that also in this case addition of the Switch-Point model reduces codelength. We conclude that the Switch-Point procedure achieved an improvement in all cases where the momentum phenomenon occurred.

However, no strong conclusions can be justified based on the properties of probabilistically generated sequences. In the next section we therefore examine the difference in codelength in expectation.

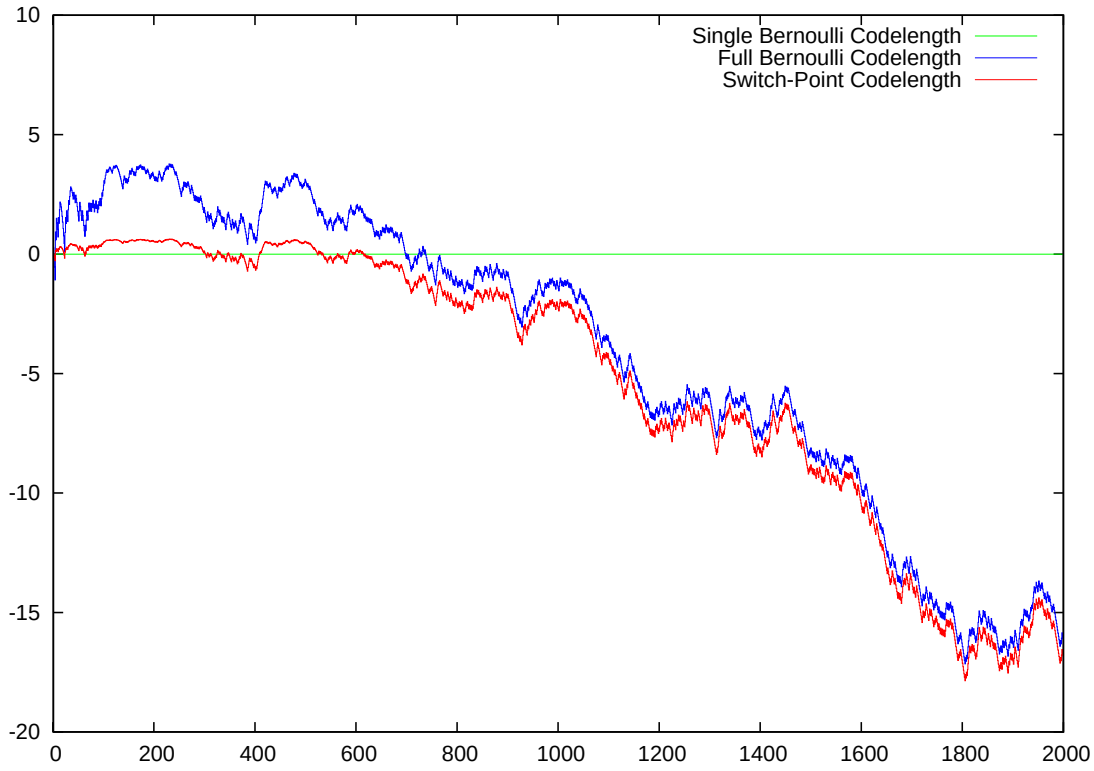


(a) $\theta^* = 0.6$

Figure 2.3: $L_1(x^n) - L_0(x^n)$ and $L_s(x^n) - L_0(x^n)$ on sequences x^n sampled from the Bernoulli distribution with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 2000$.

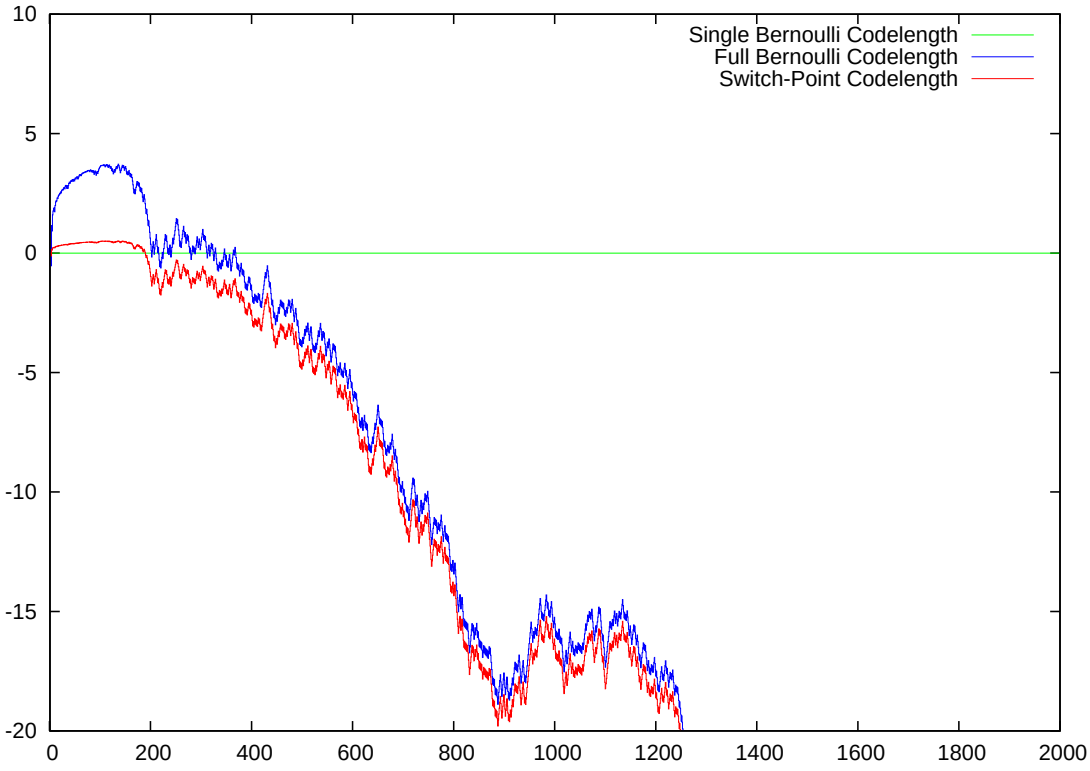


(b) $\theta^* = 0.55$

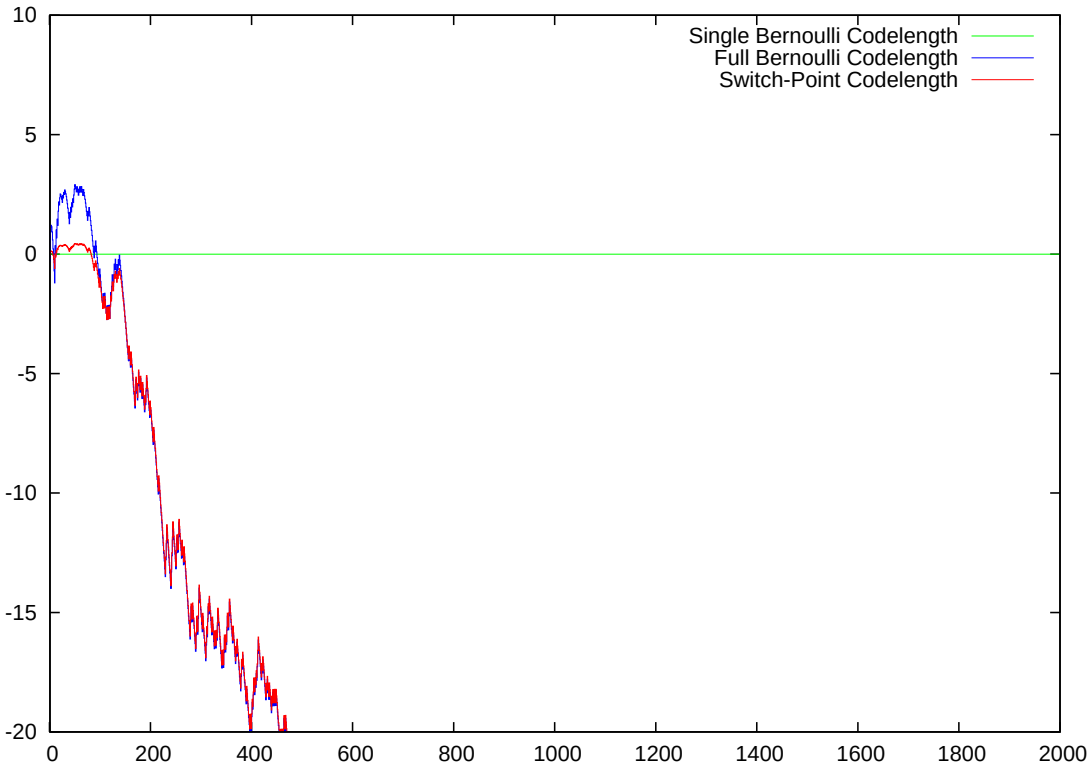


(c) $\theta^* = 0.65$

Figure 2.3 (cont.): $L_1(x^n) - L_0(x^n)$ and $L_s(x^n) - L_0(x^n)$ on sequences x^n sampled from the Bernoulli distribution with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 2000$.

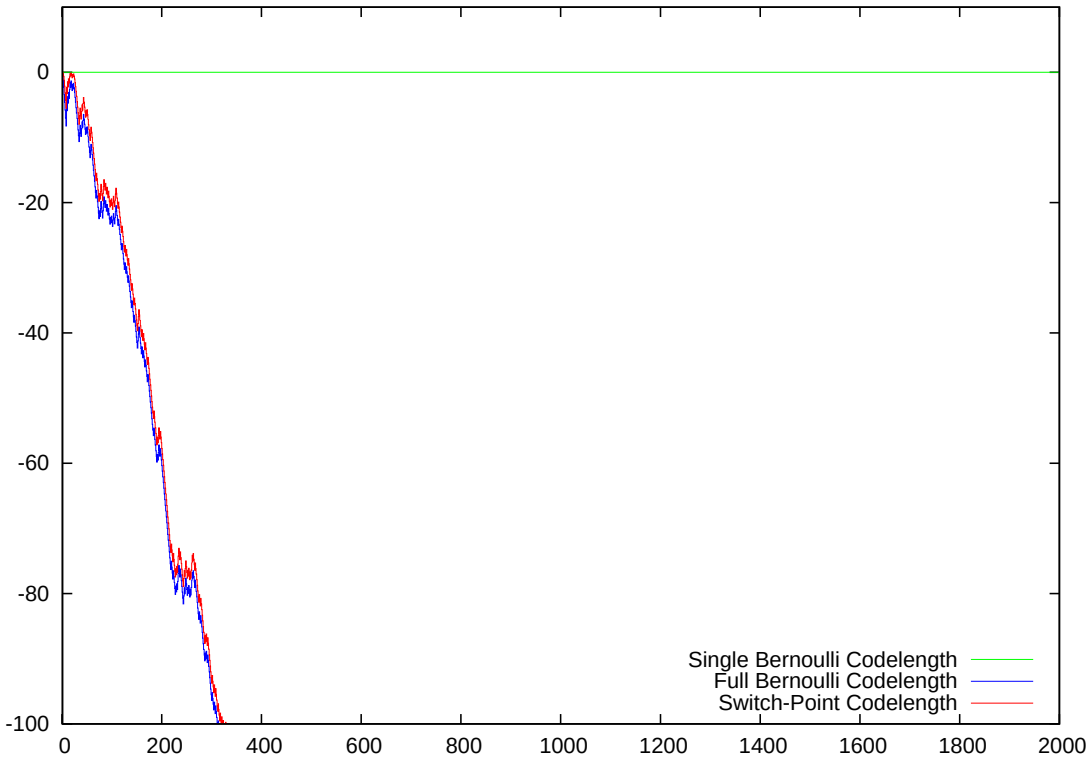


(d) $\theta^* = 0.5$

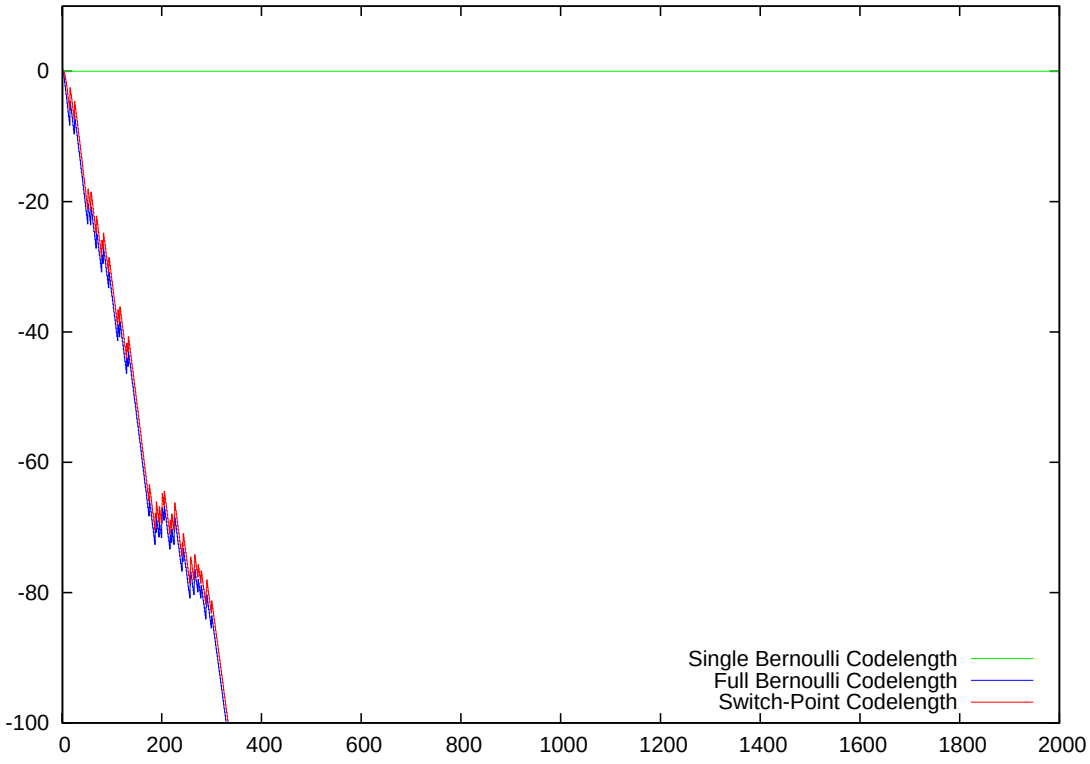


(e) $\theta^* = 0.7$

Figure 2.3 (cont.): $L_1(x^n) - L_0(x^n)$ and $L_s(x^n) - L_0(x^n)$ on sequences x^n sampled from the Bernoulli distribution with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 2000$.



(f) $\theta^* = 0.3$



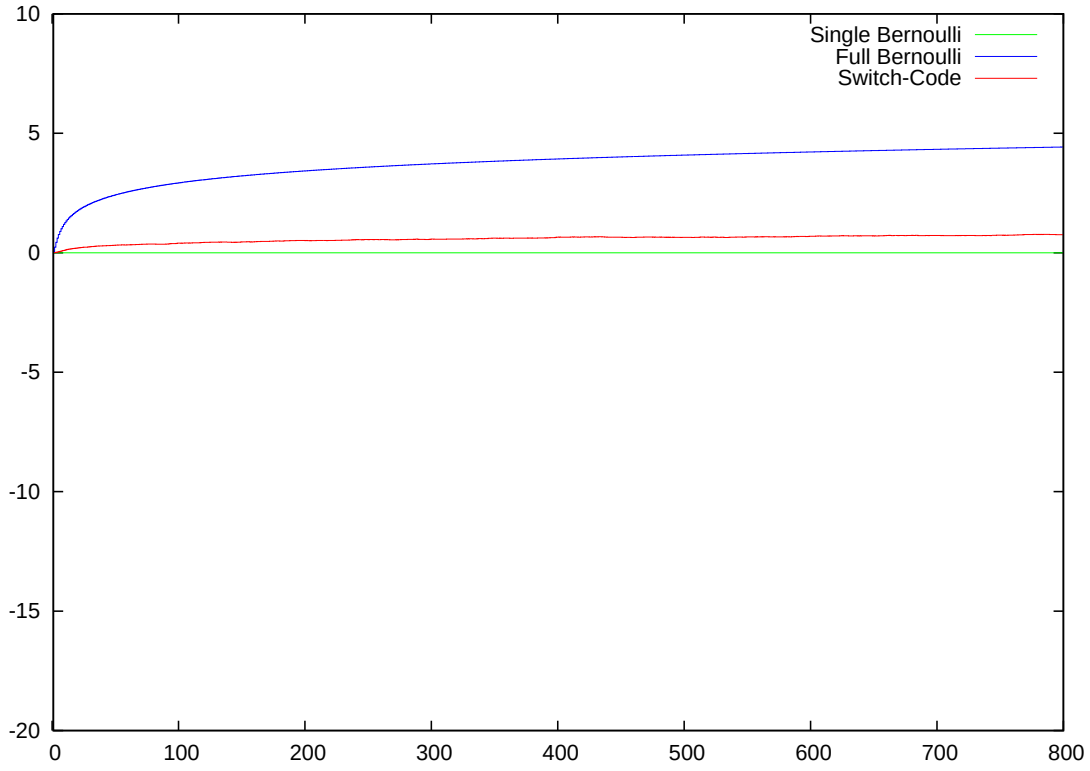
(g) $\theta^* = 0.9$

Figure 2.3 (cont.): $L_1(x^n) - L_0(x^n)$ and $L_s(x^n) - L_0(x^n)$ on sequences x^n sampled from the Bernoulli distribution with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 2000$. N.B. The scale of the vertical axis in the figures on this page differs from the scale in the previous figures!

2.3.2 Results: Expected Switch-Point Codelength

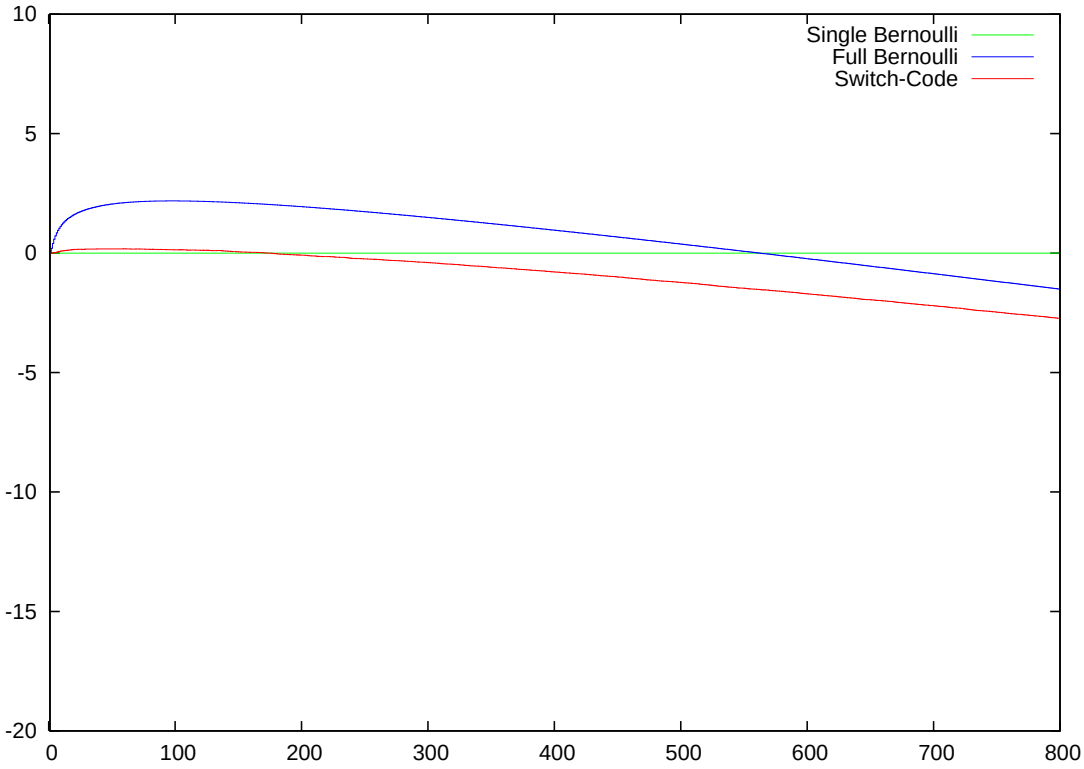
We now compare the Switch-Point codelength to the expected codelengths of the original two Bernoulli models. Figure 2.4 repeats Figure 2.2 with the expected codelength of the Switch-Point code added. For ease of comparison the expected codelength of the Single Bernoulli model has been subtracted. Due to computational limitations the expected codelength of the Switch-Point code has been approximated by averaging over 10 000 random samples from the generating distribution.

We consider again the generating distributions for which we have observed the momentum phenomenon. They are shown in Figures 2.3(b), 2.3(c), 2.3(d) and 2.3(e), which show the expected codelengths under the generating distributions in the Full Bernoulli model with θ^* equal to 0.55, 0.65, 0.5 and 0.7 respectively. We observe that in each case there exists a large range of sample sizes such that the expected Switch-Point codelength is shorter than the expected codelengths of the other two models. For these sample sizes the Switch-Point procedure achieves shorter codelength on average than MDL and Bayes. However, under the generating distributions with θ^* equal to 0.5 and 0.7 the expected codelength of the Full Bernoulli model is smaller than the Switch-Point codelength for sample sizes larger than approximately 347 and 360 respectively. We conjecture that for sufficiently large sample sizes the same will happen under the generating distributions with θ^* equal to 0.55 and 0.65. Therefore the range of sample sizes for which the Switch-Point procedure achieves shorter codelength on average than MDL and Bayes is bounded in this example. Apparently in many cases the Switch-Point code keeps assigning significant probability to the possibility that the optimal switch-point has not yet been observed. We conclude that the Switch-Point procedure in the current example (slightly) improves expected predictive performance for a range of sample sizes whenever the momentum phenomenon is likely to occur. As this suggests that it improves predictive on many individual sequences, it follows that the inefficiency of MDL and Bayes in dealing with the momentum phenomenon can be exploited. Therefore the momentum phenomenon is actually a momentum *problem*.

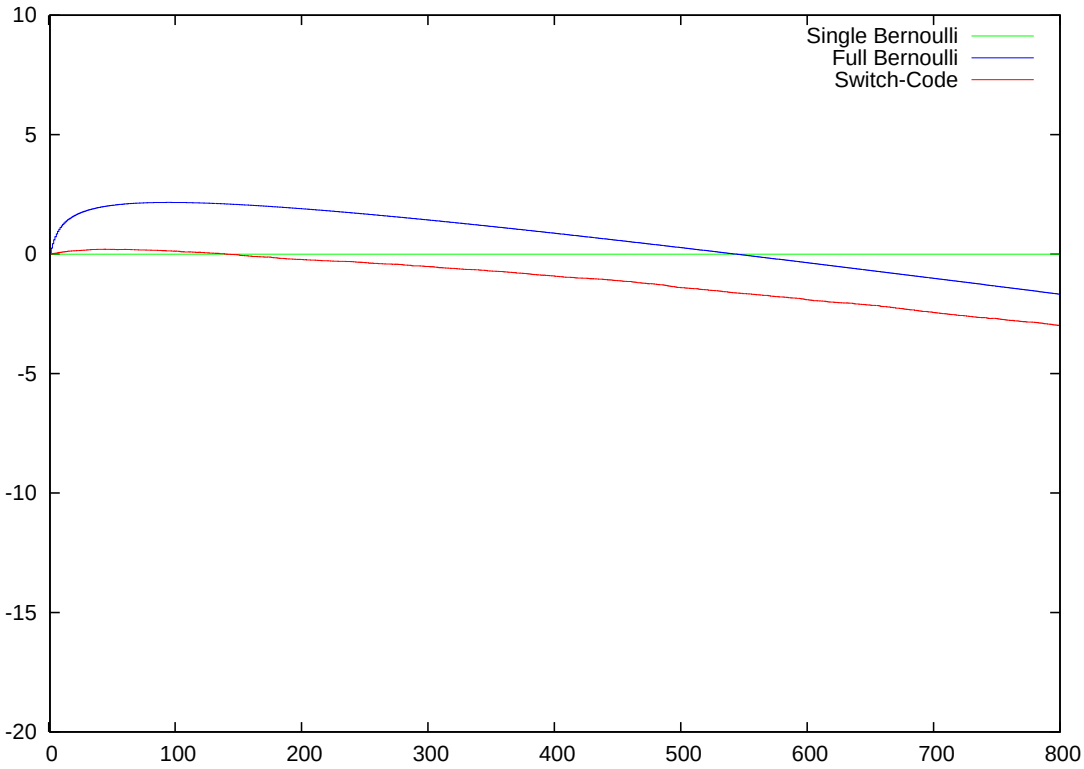


(a) $\theta^* = 0.6$

Figure 2.4: $E_{P_{\theta^*}}[L_1(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ and $E_{P_{\theta^*}}[L_{\mathrm{s}}(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ under Bernoulli distribution $P_{\theta^*} \in \mathcal{M}_1$ with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 800$. $E_{P_{\theta^*}}[L_{\mathrm{s}}(X^n)]$ has been approximated by averaging over 10 000 random samples from P_{θ^*}.

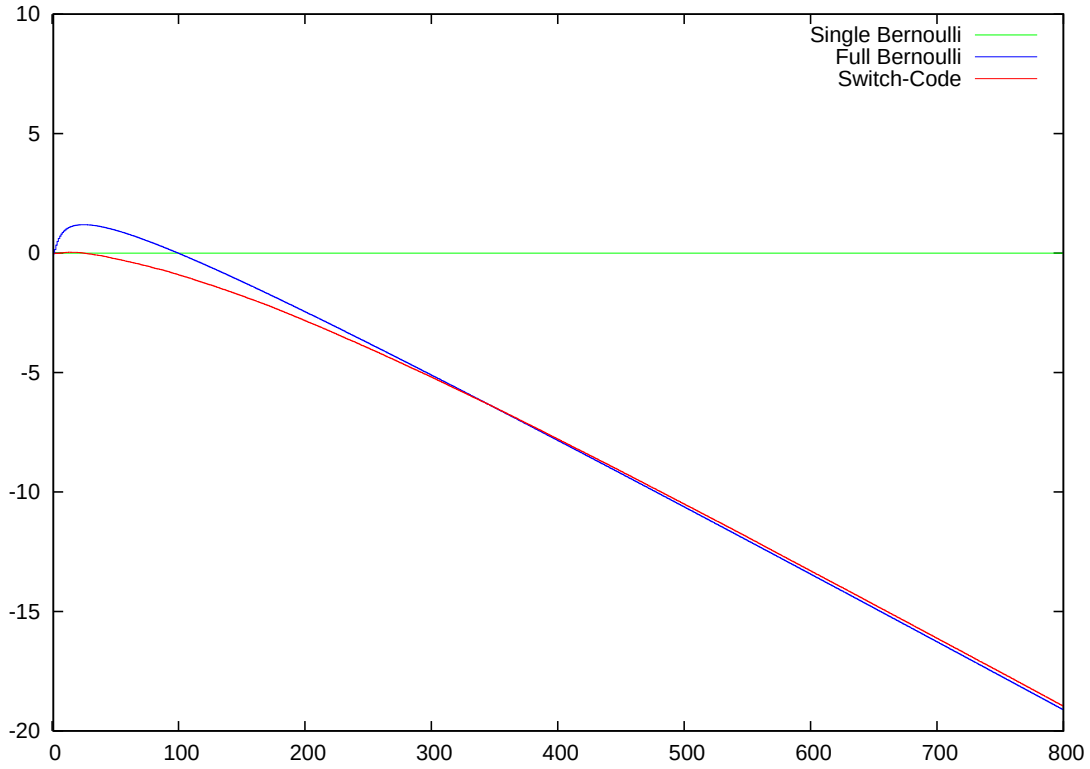


(b) $\theta^* = 0.55$

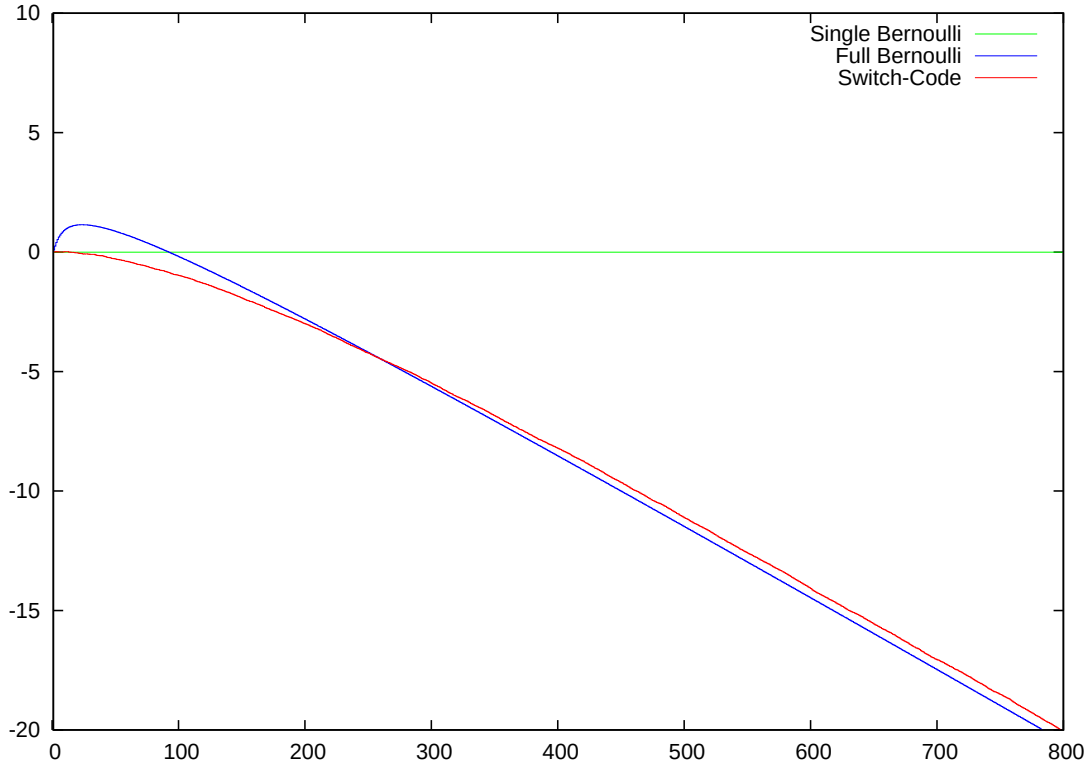


(c) $\theta^* = 0.65$

Figure 2.4 (cont.): $E_{P_{\theta^*}}[L_1(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ and $E_{P_{\theta^*}}[L_{\mathrm{s}}(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ under Bernoulli distribution $P_{\theta^*} \in \mathcal{M}_1$ with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 800$. $E_{P_{\theta^*}}[L_{\mathrm{s}}(X^n)]$ has been approximated by averaging over 10 000 random samples from P_{θ^*}.

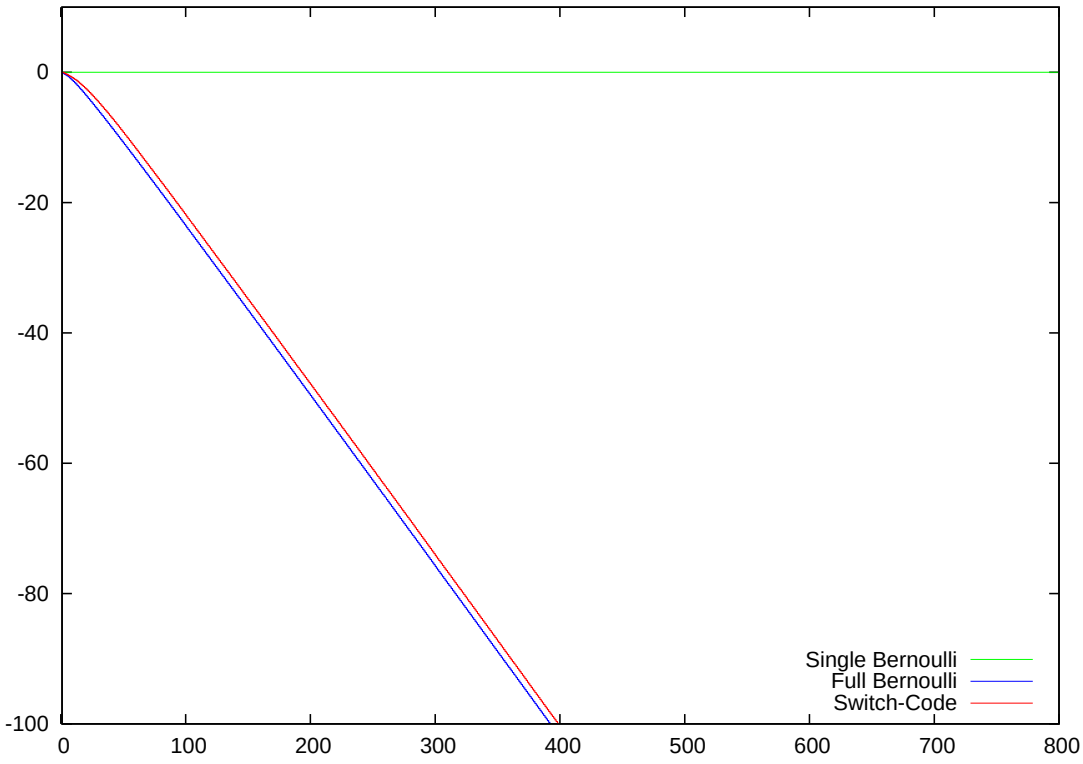


(d) $\theta^* = 0.5$

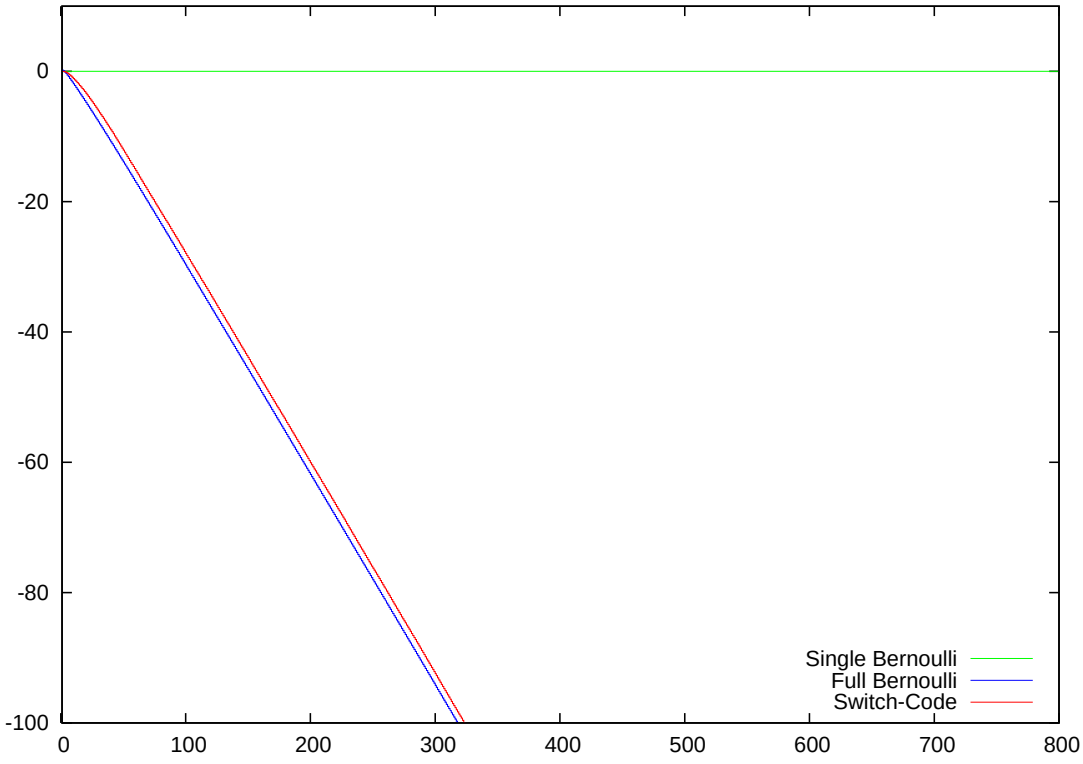


(e) $\theta^* = 0.7$

Figure 2.4 (cont.): $E_{P_{\theta^*}}[L_1(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ and $E_{P_{\theta^*}}[L_{\mathrm{s}}(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ under Bernoulli distribution $P_{\theta^*} \in \mathcal{M}_1$ with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 800$. $E_{P_{\theta^*}}[L_{\mathrm{s}}(X^n)]$ has been approximated by averaging over 10 000 random samples from P_{θ^*}.



(f) $\theta^* = 0.3$



(g) $\theta^* = 0.9$

Figure 2.4 (cont.): $E_{P_{\theta^*}}[L_1(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ and $E_{P_{\theta^*}}[L_{\mathrm{s}}(X^n)] - E_{P_{\theta^*}}[L_0(X^n)]$ under Bernoulli distribution $P_{\theta^*} \in \mathcal{M}_1$ with $P(X = 1) = \theta^*$ for $n = 1, \ldots, 800$. $E_{P_{\theta^*}}[L_{\mathrm{s}}(X^n)]$ has been approximated by averaging over 10 000 random samples from P_{θ^*}. N.B. The scale of the vertical axis in the figures on this page differs from the scale in the previous figures!

2.4 Chapter Summary

In the Bernoulli example we investigated MDL and Bayesian prediction in the presence of two nested models. We observed that the best-predicting model changed over time on typical individual sequences and in expectation under some generating sources in the larger model, which suggests that it occurs for many typical sequences. We called this the momentum phenomenon and noticed that it occurred whenever the generating source was close, but not equal, to the source in the smaller model. The momentum phenomenon has been explained as the result of the slow convergence of the best model when a quickly converging adequate model was available. The maximum number of bits that an optimal code might gain if it switches between the models at exactly the right sample size, we have called the size of the momentum phenomenon.

We then introduced the Switch-Point procedure, which adds the extra Switch-Point model to the MDL and Bayesian procedures. The Switch-Point model explicitly models at which sample size we should switch between the original models. We first proved that the Switch-Point procedure could never predict significantly worse than regular MDL and Bayes and then showed that it slightly reduced predictive loss on all the individual sequences that exhibited the momentum phenomenon. In addition, it reduced expected predictive loss on a range of sample sizes for the corresponding generating sources, which also exhibited the momentum phenomenon in expectation. We concluded that the inefficiency of MDL and Bayes in dealing with the momentum phenomenon can be exploited. Therefore the momentum phenomenon should be considered a momentum problem.

CHAPTER 3

Characteristics of the Momentum Problem

In this chapter we will prove that the momentum phenomenon can get arbitrarily large in probability for the Bernoulli example under some generating sources. We will then consider generalisations of the Bernoulli example. First, we generalise the Full Bernoulli model to any exponential family with positive dimension and substitute any model containing a single source from that exponential family for the Single Bernoulli model. For this setting we will prove that the momentum phenomenon can get arbitrarily large for expected codelength under some generating sources. Then we will consider another concrete example called the *Conditional Bernoulli example*. The Conditional Bernoulli example increases the difference in complexity between the models from the Bernoulli example by conditioning on an auxiliary variable. This increases the difference in the number of observations that are required before reasonable parameter values can be learned for the models. We will demonstrate that the Switch-Point code (and hence the Switch-Point procedure) significantly reduces predictive loss on the Conditional Bernoulli example compared to regular MDL or Bayes.

3.1 Proof: Momentum Phenomenon in Probability

In this section we state and prove Theorem 3.1.1, which shows that the momentum phenomenon may get arbitrarily large with arbitrarily high probability. We generalise the Single Bernoulli model to any nested model of the

Full Bernoulli model that contains only a single probabilistic source. To be precise, consider the event that the Full Bernoulli model loses at least C bits relative to the Single Bernoulli model on the first n_1 outcomes, but achieves shortest codelength after n_2 outcomes. Then Theorem 3.1.1 shows that this event occurs for arbitrarily large C with arbitrarily high probability for all generating sources in a set Ψ. This set Ψ depends on C and the desired probability. It is the set of sources in the Full Bernoulli model corresponding to a range in the parameter space around the parameter of the source in the Single Bernoulli model. This range shrinks when C or the desired probability are increased. We will show in Section 4.2 that the Switch-Point code will with the same probability gain at least $C - [-\log w_s(n_1)]$ bits compared to either of the two other models. As n_1 is known before observing any data, we may select w_s such that $-\log w_s(n_1)$ is small. However, the size of Ψ decreases and n_1 increases for larger values of C. Therefore in the selection of w_s there is a trade-off between the number of generating sources for which it is able to realise a significant reduction in codelength and the size of the reduction in codelength.

The difference in conditional codelength between the two models on each single outcome can be bounded. Therefore the minimum number of samples that the Full Bernoulli model needs before it catches up with the Single Bernoulli model after losing C bits, must increase with increasing C. That is, we can get the minimum number of samples at which MDL and Bayes predict suboptimally arbitrarily large by increasing C. This will not be proved formally.

Theorem 3.1.1 *Let $L_1(x^n)$ and $L_0(x^n)$ denote the codelength assigned to any data sequence x^n by the Bayesian universal models for the Full Bernoulli model, $\mathcal{M}_1$, with Jeffreys' prior and a nested model that contains only the Bernoulli probabilistic source with parameter $\theta_0 \in (0,1)$, $\mathcal{M}_0$, respectively. Then for any constant $C > 0$ and any $\epsilon > 0$ there exist a sample size n_1 and a set $\Psi := \{P_\theta \in \mathcal{M}_1 : \theta \in [1-(1-\theta_0)\cdot 2^{1/n_1}, \theta_0) \cup (\theta_0, \theta_0 \cdot 2^{1/n_1}]\} \subset \mathcal{M}_1 \setminus \mathcal{M}_0$ such that for any $P_{\theta^*} \in \Psi$ for all sufficiently large sample sizes $n_2 > n_1$*

$$P_{\theta^*}^{n_2}\Big(L_1(x^{n_1}) > L_0(x^{n_1}) + C,\ L_1(x^{n_2}) < L_0(x^{n_2})\Big) \geq 1 - \epsilon, \tag{3.1}$$

where x^{n_1} is the prefix of length n_1 of x^{n_2}.

We first prove two lemmas that are related to Theorem 3.1.1. Then we prove the theorem by combining the lemmas.

Lemma 3.1.2 (Theorem 3.1.1, Part 1) *Let $L_1(x^n)$, $L_0(x^n)$, $\mathcal{M}_1$ and $\mathcal{M}_0$ be as in Theorem 3.1.1. Then for any constant $C > 0$ and any $\epsilon_1 > 0$ there exist a sample size n_1 and a set $\Psi := \{P_\theta \in \mathcal{M}_1 : \theta \in [1-(1-\theta_0)\cdot 2^{1/n_1}, \theta_0) \cup (\theta_0, \theta_0 \cdot 2^{1/n_1}]\} \subset \mathcal{M}_1 \setminus \mathcal{M}_0$ such that for any $P_{\theta^*} \in \Psi$*

$$P_{\theta^*}^{n_1}\Big(L_1(x^{n_1}) > L_0(x^{n_1}) + C\Big) \geq 1 - \epsilon_1. \tag{3.2}$$

Proof of Lemma 3.1.2 Let functions a and b be defined as

$$a(n_1) := 1 - (1-\theta_0)\cdot 2^{1/n_1}, \quad b(n_1) := \theta_0 \cdot 2^{1/n_1}. \tag{3.3}$$

This definition will be motivated later. To exhibit an n_1 that satisfies the lemma, we will show that (3.2) is satisfied for *all* sufficiently large n_1 and all $P_{\theta^*} \in \Psi$ if Ψ is defined as

$$\Psi := \{P_\theta \in \mathcal{M}_1 : \theta \in [a(n_1), \theta_0) \cup (\theta_0, b(n_1))]\}. \tag{3.4}$$

We need to assume that n_1 is sufficiently large in order to use several asymptotic results. As $a(n_1) < \theta_0 < b(n_1)$ for all n_1 we have that Ψ is non-empty.

Let $\Theta := (0,1)$ denote the parameter space of the Full Bernoulli model and let $\hat{\theta}(x^{n_1}) := \arg\max_{\theta\in\Theta} P_\theta(x^{n_1})$ denote the maximum likelihood parameter in the Full Bernoulli model. It is easily shown by differentiation that $\hat{\theta}(x^{n_1}) = m(x^{n_1})/n_1$, where $m(x^{n_1})$ denotes the number of ones in x^{n_1}. Then by basic probability theory

$$P_{\theta^*}^{n_1}\left(L_1(x^{n_1}) > L_0(x^{n_1}) + C\right) \geq P_{\theta^*}^{n_1}\Big(L_1(x^{n_1}) > L_0(x^{n_1}) + C, \hat{\theta}(x^{n_1}) \in A\Big) \tag{3.5}$$

for any set $A \subseteq \Theta$. We let A express the requirement that $\hat{\theta}(x^{n_1})$ is bounded away from the boundary of the parameter space by a constant $\delta > 0$ that doesn't depend on n_1. Hence, A is defined as

$$A := \{\theta : \theta \in [\delta, 1-\delta]\}. \tag{3.6}$$

We prove the lemma in three stages. In stage one we will show that, for any $P_{\theta^*} \in \mathcal{M}_1 \supset \Psi$,

$$P_{\theta^*}^{n_1}\Big(L_1(x^{n_1}) > L_0(x^{n_1}) + C \mid \hat{\theta}(x^{n_1}) \in A\Big) \geq P_{\theta^*}^{n_1}\Big(D(P_{\hat{\theta}(x^{n_1})} \| P_{\theta^*}) < h(n_1) \mid \hat{\theta}(x^{n_1}) \in A\Big) \tag{3.7}$$

for some appropriately defined function h that doesn't depend on P_{θ^*}. Here $D(\cdot\|\cdot)$ denotes Kullback-Leibler divergence. (3.7) implies that

$$P_{\theta^*}^{n_1}\left(L_1(x^{n_1}) > L_0(x^{n_1}) + C, \hat{\theta}(x^{n_1}) \in A\right) \geq P_{\theta^*}^{n_1}\left(D(P_{\hat{\theta}(x^{n_1})}\|P_{\theta^*}) < h(n_1), \hat{\theta}(x^{n_1}) \in A\right). \quad (3.8)$$

For any $\epsilon_{11}, \epsilon_{12} > 0$ we will show that for any $P_{\theta^*} \in \Psi$

$$P_{\theta^*}^{n_1}\left(\hat{\theta}(x^{n_1}) \in A\right) \geq 1 - \epsilon_{11} \quad (3.9)$$

and

$$P_{\theta^*}^{n_1}\left(D(P_{\hat{\theta}(x^{n_1})}\|P_{\theta^*}) < h(n_1)\right) \geq 1 - \epsilon_{12} \quad (3.10)$$

for all sufficiently large n_1. Selecting ϵ_{11} and ϵ_{12} such that $\epsilon_{11} + \epsilon_{12} \leq \epsilon_1$, Equations 3.9 and 3.10 imply that

$$P_{\theta^*}^{n_1}\left(D(P_{\hat{\theta}(x^{n_1})}\|P_{\theta^*}) < h(n_1), \hat{\theta}(x^{n_1}) \in A\right) \geq 1 - \epsilon_1 \quad (3.11)$$

by basic probability theory. We will prove (3.9) in stage two and (3.10) in stage three. Tracing back our steps by substituting (3.11) into (3.8) and the result into (3.5), then completes the proof of the lemma.

Stage One We start by proving (3.7). We first find a lower bound for $L_1(x^{n_1})$ and an upper bound for $L_0(x^{n_1})$. We expand $L_1(x^{n_1})$ as

$$L_1(x^{n_1}) = -\log P_{\hat{\theta}(x^{n_1})}(x^{n_1}) + \frac{1}{2}\log\frac{n_1}{2\pi} + \log\frac{\sqrt{|I(\hat{\theta})|}}{w(\hat{\theta})} + f\left(\hat{\theta}(x^{n_1}), n_1\right) \quad (3.12)$$

$$= -\log P_{\hat{\theta}(x^{n_1})}(x^{n_1}) + \frac{1}{2}\log\frac{n_1}{2\pi} + \log\int\sqrt{|I(\theta)|}\,d\theta + f\left(\hat{\theta}(x^{n_1}), n_1\right), \quad (3.13)$$

where $f : \Theta \times \mathbb{N} \to \mathbb{R}$ is of order $o(1)$ for fixed $\theta \in \Theta$. The expansion holds for the Full Bernoulli model if $\hat{\theta}(x^{n_1}) \in A$ [Grünwald et al., 2005; Balasubramanian, 1997]. It will prove useful to replace $f(\hat{\theta}(x^{n_1}), n_1)$ by a smaller term that doesn't depend on $\hat{\theta}(x^{n_1})$. We do this by minimising over all $\theta \in A$ and replace $f(\hat{\theta}(x^{n_1}), n_1)$ by

$$f(n_1) := \min_{\theta \in A} f(\theta, n_1). \quad (3.14)$$

It follows from [Balasubramanian, 1997] that $f(n_1)$ is of order $o(1)$, which implies $O(1)$. For the Full Bernoulli model $\log \int_{\theta\in\Theta} \sqrt{|I(\theta)|}\, d\theta = \log \pi = O(1)$. Grouping $O(1)$ terms in (3.13) together, we therefore get that

$$L_1(x^{n_1}) \geq -\log P_{\hat{\theta}(x^{n_1})}(x^{n_1}) + \frac{1}{2}\log n_1 + C' \tag{3.15}$$

for some constant C'.

Regarding $L_0(x^{n_1})$, it holds for any $P_{\theta^*} \in \Psi$ that

$$\max_{x^{n_1}} \{L_0(x^{n_1}) - [-\log P_{\theta^*}(x^{n_1})]\} = n_1 \cdot \max_x \{-\log P_{\mathcal{M}_0}(x) + \log P_{\theta^*}(x)\} \tag{3.16}$$

$$= n_1 \cdot \max_x \left\{\log \frac{P_{\theta^*}(x)}{P_{\mathcal{M}_0}(x)}\right\} \tag{3.17}$$

$$= n_1 \cdot \log\left(\max\left\{\frac{\theta^*}{\theta_0}, \frac{1-\theta^*}{1-\theta_0}\right\}\right). \tag{3.18}$$

For any distribution $P_{\theta^*} \in \Psi$, we have that $\theta^* \in [a(n_1), \theta_0) \cup (\theta_0, b(n_1)]$. Therefore the maximum in (3.18) depends on $a(n_1)$ and $b(n_1)$. The definitions of a and b have been chosen such that for all $P_{\theta^*} \in \Psi$

$$L_0(x^{n_1}) \leq -\log P_{\theta^*}(x^{n_1}) + 1. \tag{3.19}$$

for any x^{n_1}.

By (3.15) and (3.19) we get that for some constant C''

$$P_{\theta^*}^{n_1}\left(L_1(x^{n_1}) > L_0(x^{n_1}) + C \mid \hat{\theta}(x^{n_1} \in A\right) \geq$$
$$P_{\theta^*}^{n_1}\left(-\log P_{\hat{\theta}(x^{n_1})}(x^{n_1}) + \frac{1}{2}\log n_1 > -\log P_{\theta^*}(x^{n_1}) + C'' \mid \hat{\theta}(x^{n_1}) \in A\right). \tag{3.20}$$

Rewriting gives

$$P_{\theta^*}^{n_1}\left(L_1(x^{n_1}) > L_0(x^{n_1}) + C \mid \hat{\theta}(x^{n_1} \in A\right) \geq$$
$$P_{\theta^*}^{n_1}\left(\log \frac{P_{\hat{\theta}(x^{n_1})}(x^{n_1})}{P_{\theta^*}(x^{n_1})} < \frac{1}{2}\log n_1 + C'' \mid \hat{\theta}(x^{n_1}) \in A\right). \tag{3.21}$$

Letting $m(x^{n_1})$ denote the number of ones in x^{n_1}, we have that

$$-\log P_\theta(x^{n_1}) = m(x^{n_1}) \cdot -\log\theta + (n_1 - m(x^{n_1})) \cdot -\log(1-\theta) \tag{3.22}$$

$$= n_1 E_{\mathbb{P}_{x^{n_1}}}[-\log P_\theta(X)] \tag{3.23}$$

for all $P_\theta \in \mathcal{M}_1$, all n_1 and all x^{n_1}. This equivalence holds for all exponential families that are extended to multiple outcomes by taking product distributions [Grünwald, 2007]. In addition, for the Bernoulli model the empirical distribution $\mathbb{P}_{x^{n_1}}$ is equal to the maximum likelihood distribution $P_{\hat\theta(x^{n_1})}$ if $\hat\theta(x^{n_1})$ exists in $\Theta \supset A$. Hence, conditional on $\hat\theta(x^{n_1}) \in A$,

$$\log \frac{P_{\hat\theta(x^{n_1})}(x^{n_1})}{P_{\theta^*}(x^{n_1})} = n_1 E_{\mathbb{P}_{x^{n_1}}}\left[\log \frac{P_{\hat\theta(x^{n_1})}(X)}{P_{\theta^*}(X)}\right] \tag{3.24}$$

$$= n_1 E_{P_{\hat\theta(x^{n_1})}}\left[\log \frac{P_{\hat\theta(x^{n_1})}(X)}{P_{\theta^*}(X)}\right] \tag{3.25}$$

$$= n_1 D(P_{\hat\theta(x^{n_1})} \| P_{\theta^*}). \tag{3.26}$$

Substitution into (3.21) gives

$$P_{\theta^*}^{n_1}\left(L_1(x^{n_1}) > L_0(x^{n_1}) + C \mid \hat\theta(x^{n_1}) \in A\right) \geq$$

$$P_{\theta^*}^{n_1}\left(D(P_{\hat\theta(x^{n_1})} \| P_{\theta^*}) < \frac{\frac{1}{2}\log n_1 + C''}{n_1} \mid \hat\theta(x^{n_1}) \in A\right). \tag{3.27}$$

Letting

$$h(n_1) := \frac{\frac{1}{2}\log n_1 + C''}{n_1}, \tag{3.28}$$

now completes the proof of (3.7).

Stage Two We will now prove (3.9). We choose δ sufficiently small such that $\delta < a(1) \leq a(n_1)$ and $1 - \delta > b(1) \geq b(n_1)$. For any fixed $P_{\theta^*} \in \Psi$ and any constant $\gamma > 0$ we have by the law of large numbers that

$$P_{\theta^*}(|\hat\theta(x^{n_1}) - \theta^*| \leq \gamma) \to 1. \tag{3.29}$$

By taking γ sufficiently small, we have for all n_1 that

$$\{x^{n_1} : |\hat\theta(x^{n_1}) - \theta^*| \leq \gamma\} \subseteq \{x^{n_1} : \hat\theta(x^{n_1}) \in A\}. \tag{3.30}$$

Therefore

$$P_{\theta^*}(\hat\theta(x^{n_1}) \in A) \to 1, \tag{3.31}$$

which implies (3.9).

Stage Three Our next step is to prove (3.10). We will show that the probability of the converse of (3.10) does not exceed ϵ_{12} for sufficiently large n_1. In the proof we will use the following theorem. Suppose E is a convex set of probability distributions on $\mathcal{X}$ and define $P(E) := P(\{x^n : \mathbb{P}_{x^n} \in E\})$. Then, for any distribution P,

$$P(E) \leq 2^{-nD^{\min}} \tag{3.32}$$

if $D^{\min} := \min_{P' \in E} D(P'\|P)$ exists[Grünwald, 2007]. A similar result, which applies also if the minimum in the definition of $D^{\min}$ is replaced by an infimum, is obtained by Csiszár [1984, Equation 2.16]. This latter result, however, requires the stronger condition that E be almost completely convex, which I have been unable to verify in the applications of the theorem below.

We will also use several properties of the Bernoulli model on single outcomes that hold for all exponential families if every distribution is indexed by its mean, which is called the *mean-value parameterisation* [Grünwald, 2007]. Hence, we would reparameterise the Bernoulli model by its mean-value parameterisation. This is unnecessary, however, as the common parameterisation for the Bernoulli model, which we have used until now, already indexes each distribution over a single outcome by its mean:

$$E_\theta[X] = P_\theta(X = 1) = \theta. \tag{3.33}$$

For the proof, let E_1 and E_2 be defined as

$$E_1 := \{P_\theta \in \mathcal{M}_1 : D(P_\theta\|P_{\theta^*}) \geq h(n_1), \theta \leq \theta^*\}, \tag{3.34}$$
$$E_2 := \{P_\theta \in \mathcal{M}_1 : D(P_\theta\|P_{\theta^*}) \geq h(n_1), \theta \geq \theta^*\}. \tag{3.35}$$

We will apply (3.32) to E_1 and E_2. Convexity of E_1 and E_2 will now be proved simultaneously. Let $i = 1, 2$. For exponential families in their mean-value parameterisation, $D(P_\theta\|P_{\theta^*})$ is a convex function of θ. In addition, it follows from the definition of Kullback-Leibler divergence that $D(P_\theta\|P_{\theta^*}) = 0$ iff $\theta = \theta^*$. Therefore the region in the parameter space corresponding to E_i must be convex.

We need to show that any distribution P that is a linear combination of any distributions $P_{\theta'}$ and $P_{\theta''}$ in E_i is also in E_i. Without loss of generality we will assume that $\theta' \leq \theta''$. Let P be given by

$$P(x) = \alpha P_{\theta'}(x) + (1 - \alpha)P_{\theta''}(x) \tag{3.36}$$

for any arbitrary $P_{\theta'}, P_{\theta''} \in E_i$ and any arbitrary α such that $0 \leq \alpha \leq 1$. The only possible distributions on binary outcome spaces are Bernoulli distributions. Therefore P is a Bernoulli distribution with some mean μ. We now show that $\theta' \leq \mu \leq \theta''$ and therefore that $P \in E_i$:

$$\begin{aligned}\mu &= E_P[X] && (3.37)\\ &= \sum_x \alpha \cdot P_{\theta'}(x) + (1-\alpha) \cdot P_{\theta''}(x) && (3.38)\\ &= \alpha E_{P_{\theta'}}[X] + (1-\alpha) E_{P_{\theta''}}[X] && (3.39)\\ &= \alpha \cdot \theta' + (1-\alpha) \cdot \theta''. && (3.40)\end{aligned}$$

Let $i = 1, 2$. Convexity of $D(P_\theta \| P_{\theta^*})$ in θ implies continuity. Therefore $D_i^{\min} = \min_{P_\theta \in E_i} D(P_\theta \| P_{\theta^*}) = h(n_1)$ exists if E_i is not empty, and we can apply (3.32) to get an upper bound on the probability of E_i. Thus

$$\begin{aligned}P_{\theta^*}(E_i) &\leq 2^{-n_1 h(n_1)} && (3.41)\\ &= 2^{-n_1 \frac{\log \sqrt{n_1} + C''}{n_1}} && (3.42)\\ &= \frac{2^{-C''}}{\sqrt{n_1}}. && (3.43)\end{aligned}$$

In addition, if E_i is empty, then

$$P_{\theta^*}(E_i) = 0 \leq \frac{2^{-C''}}{\sqrt{n_1}}. \tag{3.44}$$

The maximum likelihood distribution may not exist for some x^{n_1}. This is the case if x^{n_1} is a sequence of either all zeroes or of all ones. The probability of this event is

$$P_{\theta^*}(\hat{\theta}(x^{n_1}) \text{ does not exist}) = \theta^{*n_1} + (1-\theta^*)^{n_1}. \tag{3.45}$$

We can now bound the probability in (3.10) from below by

$$\begin{aligned}P_{\theta^*}^{n_1} \left(D(P_{\hat{\theta}(x^{n_1})} \| P_{\theta^*}) < h(n_1) \right) \geq \\ 1 - \left(P_{\theta^*}(E_1) + P_{\theta^*}(E_2) + P_{\theta^*}(\hat{\theta}(x^{n_1}) \text{ does not exist}) \right). \end{aligned} \tag{3.46}$$

Recall that $0 < \theta^* < 1$. Therefore for all sufficiently large n_1

$$P_{\theta^*}(E_1) + P_{\theta^*}(E_2) + P_{\theta^*}(\hat{\theta}(x^{n_1}) \text{ does not exist}) \leq$$
$$2 \cdot \frac{2^{-C''}}{\sqrt{n_1}} + \theta^{*n_1} + (1-\theta^*)^{n_1} \leq \epsilon_{12}, \quad (3.47)$$

which completes the proof of (3.10) and thereby the proof of the lemma.

□

Lemma 3.1.3 (Theorem 3.1.1, Part 2) *Let $L_1(x^n)$, $L_0(x^n)$, $\mathcal{M}_1$ and $\mathcal{M}_0$ be as in Theorem 3.1.1. Then, for any $\epsilon_2 > 0$, any sample size n_1 and any set $\Psi \subseteq \mathcal{M}_1 \setminus \mathcal{M}_0$, for any $P_{\theta^*} \in \Psi$ for all sufficiently large sample sizes $n_2 > n_1$*

$$P_{\theta^*}^{n_2}\Big(L_1(x^{n_2}) < L_0(x^{n_2})\Big) \geq 1 - \epsilon_2. \quad (3.48)$$

Proof of Lemma 3.1.3 Intuitively, (3.48) expresses consistency of MDL (see Section 1.4.4). Formally, as $\Psi \subseteq \mathcal{M}_1 \setminus \mathcal{M}_0$, we use that for all $P_{\theta^*} \in \Psi$

$$P_{\theta^*}^{n_2}\Big(L_1(x^{n_2}) < L_0(x^{n_2})\Big) \to 1 \quad (3.49)$$

as $n_2 \to \infty$, which implies (3.48). Although, technically, $\mathcal{M}_0$ is not an exponential family, it is readily seen that the arguments in [Barron et al., 1998] that show (3.49), among others, for nested exponential families, transfer directly.

□

Proof of Theorem 3.1.1 (Combining Lemma 3.1.2 and 3.1.3) By compatibility of probabilistic sources (see page 4), (3.2) implies that

$$P_{\theta^*}^{n_2}\left(L_1(x^{n_1}) > L_0(x^{n_1}) + C\right) \geq 1 - \epsilon_1. \quad (3.50)$$

We select $\epsilon_1, \epsilon_2 > 0$ such that $\epsilon_1 + \epsilon_2 \leq \epsilon$. Then, by basic probability theory, (3.50) and (3.48) together imply that

$$P_{\theta^*}^{n_2}\left(L_1(x^{n_1}) > L_0(x^{n_1}) + C,\ L_1(x^{n_2}) < L_0(x^{n_2})\right) \geq 1 - \epsilon, \quad (3.51)$$

which completes the proof of Theorem 3.1.1.

□

3.2 Proof: Momentum Phenomenon in Expectation

In the previous section we showed that, in the Bernoulli example, the momentum phenomenon may get very large with high probability. In this section we generalise the models in the Bernoulli example to two models $\mathcal{M}_a$ and $\mathcal{M}_b$ and summarise the difference in codelength between the codes for $\mathcal{M}_a$ and $\mathcal{M}_b$ by its expected value instead. Let $L_a(x^n)$ and $L_b(x^n)$ denote the codelength assigned to any data sequence x^n by the codes for $\mathcal{M}_a$ and $\mathcal{M}_b$, respectively. Then we say that the momentum phenomenon occurs for expected codelength if there exist sample sizes n_1, n_2 with $n_1 < n_2$ such that $E[L_b(X^{n_1}) - L_a(X^{n_1})] > C$ and $E[L_b(X^{n_2}) - L_a(X^{n_2})] < 0$ for some constant $C > 0$.[1] We call the maximum C for which the expected codelength exhibits the momentum phenomenon the *size of the momentum phenomenon for expected codelength*. We will state and prove Theorem 3.2.1, which shows that the momentum phenomenon for expected codelength may get very large.

Model $\mathcal{M}_b$ is a generalisation of the Full Bernoulli model to any discrete exponential family with finite dimension and model $\mathcal{M}_a$ generalises the Single Bernoulli model to any model containing a single source from $\mathcal{M}_b$. We call an exponential family discrete if it is defined on a countable outcome space. In addition, we generalise Jeffreys' prior to any continuous positive prior on the parameter space Θ of $\mathcal{M}_b$. For exponential families the Fisher information matrix $I(\theta)$ (defined in (1.11)) is always positive definite and the likelihood functions $\ln P_\theta(x)$ are infinitely differentiable to θ [Barndorff-Nielsen, 1978]. As a consequence $|I(\theta)| > 0$ and $I(\theta)$ is continuous in θ. Therefore Jeffreys' prior, if it exists, is always continuous and positive for exponential families, although for some exponential families it does not exist. For such exponential families the integral, $\int \sqrt{|I(\theta)|}\, d\theta$, in its definition diverges.

Following Clarke and Barron [1990] we define a d-dimensional exponential family on finite or countably infinite outcome space $\mathcal{X}$ as a set of probability distributions of the form

$$P_\theta(x) := e^{-\theta^T \phi(x)} g(x)/c(\theta)$$

with natural parameter space $\Theta = \{\theta \in \mathbb{R}^d : c(\theta) < \infty\}$, where $c(\theta) = \int e^{-\theta^T \phi(x)} g(x)\, dx$ is a normalising constant and the function $g(x) : \mathcal{X} \to \mathbb{R}$ is an arbitrary nonnegative function that is positive for at least one $x \in \mathcal{X}$. In addition, we assume that the vector-valued function $\phi(x)$ is such that

[1] Compare the definition of the momentum phenomenon on page 28.

$\theta^T \phi(x)$ is a non-constant function of x, except for $\theta = \mathbf{0}$. This implies that the dimensionality of the family cannot be reduced [Barndorff-Nielsen, 1978]. Examples of probability distributions that can be written in this form are the Poisson, geometric, Bernoulli and multinomial distributions. The distributions in the exponential family are extended to n outcomes by taking the n-fold product distribution, such that

$$P_\theta^n(x^n) := \prod_{i=1}^{n} P_\theta(x_i).$$

As in the proof of Theorem 3.1.1 in the previous section, it will be shown in Theorem 3.2.1 that the size of the momentum phenomenon for expected codelength exceeds an arbitrarily large constant $C > 0$ for all generating sources in a set Ψ. As in the previous section, the set Ψ consists of the sources in $\mathcal{M}_b$ corresponding to a region in the parameter space around the source in model $\mathcal{M}_a$ that shrinks if C is increased.

Theorem 3.2.1 *Let $\mathcal{M}_b := \{P_\theta : \theta \in \Theta \subseteq \mathbb{R}^d\}$ be a discrete d-dimensional exponential family in its natural parameterisation that is extended to multiple outcomes by taking product distributions, with Θ an open and convex set. Let $P_{\mathcal{M}_b}^n(x^n) := \int P_\theta^n(x^n) w(\theta)\, d\theta$ denote the probability assigned to any data sequence x^n by the Bayesian universal model $P_{\mathcal{M}_b}^n$ for $\mathcal{M}_b$ under some prior w on Θ and let $L_b(x^n) := -\log P_{\mathcal{M}_b}^n(x^n)$ denote the corresponding codelength. In addition, pick any $P_{\theta_0} \in \mathcal{M}_b$ and let $L_a(x^n) := -\log P_{\theta_0}^n(x^n)$ denote the corresponding codelength for x^n. Then for every w that is both continuous and positive everywhere on Θ and for every constant $C > 0$ there exist a sample size n_1 and a set $\Psi \subseteq \mathcal{M}_b \setminus \{P_{\theta_0}\}$ that depends on n_1 such that for all $P_{\theta^*} \in \Psi$ for all sufficiently large sample sizes $n_2 > n_1$*

$$E_{P_{\theta^*}^{n_2}}[L_b(X^{n_1}) - L_a(X^{n_1})] > C, \tag{3.52}$$

but

$$E_{P_{\theta^*}^{n_2}}[L_b(X^{n_2}) - L_a(X^{n_2})] < 0, \tag{3.53}$$

where x^{n_1} is the prefix of length n_1 of x^{n_2}.

Proof Under conditions that hold for any exponential family $\mathcal{M}_b$ and any continuous and positive prior $w(\theta)$ on the parameter space Θ of $\mathcal{M}_b$, Clarke

and Barron [1990] establish the following approximation of the Kullback-Leibler divergence $D(P^n_{\theta^*}\|P^n_{\mathcal{M}_b})$ for any $P_{\theta^*} \in \mathcal{M}_b$:

$$D(P^n_{\theta^*}\|P^n_{\mathcal{M}_b}) = \frac{d}{2}\log\frac{n}{2\pi e} + \log\frac{\sqrt{|I(\theta^*)|}}{w(\theta^*)} + f(\theta^*, n), \tag{3.54}$$

where $f(\theta^*, n)$ is of order $o(1)$ and converges uniformly to zero when n goes to infinity for θ^* in any compact set $\Upsilon \subset \Theta$. In other words, there exists a function $g(n)$ of order $o(1)$ such that $|f(\theta^*, n)| \leq g(n)$ for all $\theta^* \in \Upsilon$ and all n, where $|f(\theta^*, n)|$ denotes the absolute value of $f(\theta^*, n)$. In addition, as $0 < w(\theta) < \infty$ for all $\theta \in \Theta$ by continuity and positivity of w and, for exponential families, $0 < |I(\theta)| < \infty$ for all $\theta \in \Theta$ [Barndorff-Nielsen, 1978], we have that $\log\sqrt{|I(\theta^*)|}/w(\theta^*)$ is bounded as a function of θ^* over any compact subset of Θ. Therefore (3.54) implies that for any compact subset $\Upsilon \subset \Theta$ there must exist constants C' and C'' such that, for all n, for all $\theta^* \in \Upsilon$

$$C' \leq D(P^n_{\theta^*}\|P^n_{\mathcal{M}_b}) - \frac{d}{2}\log\frac{n}{2\pi e} \leq C''. \tag{3.55}$$

In particular, let $\Upsilon \subset \Theta$ be any closed set with non-empty interior that contains θ_0. As $\Theta \ni \theta_0$ is open, convex and bounded by assumption and definition, such a set Υ must always exist and be compact. Letting C' and C'' be constants such that (3.55) holds for all $\theta^* \in \Upsilon$, we now pick n_1 large enough such that

$$\frac{d}{2}\log\frac{n_1}{2\pi e} + C' - 1 > C, \tag{3.56}$$

and define Ψ as

$$\Psi := \left\{ P_\theta \in \mathcal{M}_b : \theta \in \Upsilon \setminus \{\theta_0\}, D(P_\theta\|P_{\theta_0}) < \frac{1}{n_1} \right\}. \tag{3.57}$$

We have the following properties, which together ensure that Ψ is non-empty. Firstly, Θ is open. Therefore θ_0 cannot lie at the boundary of Θ. Secondly, for exponential families in their natural parameterisation, $D(P_\theta\|P_{\theta_0})$ is a continuous function of θ [Grünwald, 2007] and reaches its unique minimum $D(P_\theta\|P_{\theta_0}) = 0$ at $\theta = \theta_0$. These latter two properties, together with openness of Θ, ensure that $\{\theta \in \Theta : D(P_\theta\|P_{\theta_0}) < 1/n_1\}$ is an open superset of $\{\theta_0\}$. Hence, both Υ and $\{\theta \in \Theta : D(P_\theta\|P_{\theta_0}) < 1/n_1\}$ are proper supersets of $\{\theta_0\}$ with non-empty interior. It follows that their intersection must also be a proper superset of $\{\theta_0\}$. Therefore Ψ is not empty.

We will now first prove (3.52) and then prove (3.53). As $D(P_{\theta^*}\|P_{\theta_0}) \leq 1/n_1$ for all $P_{\theta^*} \in \Psi$, we have that

$$E_{P_{\theta^*}^{n_2}}[L_b(X^{n_1}) - L_a(X^{n_1})] = E_{P_{\theta^*}^{n_1}}[L_b(X^{n_1}) - L_a(X^{n_1})] \tag{3.58}$$

$$= D(P_{\theta^*}^{n_1}\|P_{\mathcal{M}_b}^{n_1}) - n_1 D(P_{\theta^*}\|P_{\theta_0}) \tag{3.59}$$

$$\geq D(P_{\theta^*}^{n_1}\|P_{\mathcal{M}_b}^{n_1}) - 1 \tag{3.60}$$

for all $P_{\theta^*} \in \Psi$. Moreover, by (3.55) and (3.56) it follows that

$$E_{P_{\theta^*}^{n_2}}[L_b(X^{n_1}) - L_a(X^{n_1})] \geq \frac{d}{2}\log\frac{n_1}{2\pi e} + C' - 1 \tag{3.61}$$

$$\geq C, \tag{3.62}$$

which is (3.52).

To guide our intuition regarding (3.53), we note that it basically expresses consistency of MDL (see Section 1.4.4) in expectation. Formally, we use that

$$E_{P_{\theta^*}^{n_2}}[L_b(X^{n_2}) - L_a(X^{n_2})] = D(P_{\theta^*}^{n_2}\|P_{\mathcal{M}_b}^{n_2}) - n_2 D(P_{\theta^*}\|P_{\theta_0}). \tag{3.63}$$

By (3.55) it is seen that $D(P_{\theta^*}^{n_2}\|P_{\mathcal{M}_b}^{n_2})$ grows logarithmically in n_2 for sufficiently large n_2 for all $P_{\theta^*} \in \Psi$. However, $D(P_{\theta^*}\|P_{\theta_0})$ is a positive constant for all $P_{\theta^*} \in \Psi$, which implies that $n_2 D(P_{\theta^*}\|P_{\theta_0})$ grows linearly in n_2. Therefore for all $P_{\theta^*} \in \Psi$ for all sufficiently large $n_2 > n_1$

$$E_{P_{\theta^*}^{n_2}}[L_b(X^{n_2}) - L_a(X^{n_2})] < 0, \tag{3.64}$$

which is (3.53). This completes the proof of the theorem. □

3.3 Conditional Bernoulli Example

In this section we present another example, called the Conditional Bernoulli example, that exhibits the momentum problem. The Conditional Bernoulli example is a generalisation of the Bernoulli example from Section 2.1. It increases the difference in complexity between the Full Bernoulli model and the Single Bernoulli model by conditioning on an auxiliary variable. As a result the new Full Bernoulli model requires even more observations before it starts predicting as well as the Single Bernoulli model. This magnifies the momentum phenomenon, which allows us investigate it more easily. The Conditional Bernoulli example will provide a concrete example where the Switch-Point procedure gains a significant number of bits compared to MDL and Bayes due to the momentum problem.

3.3.1 Model Specification

Suppose the data consist of paired observations (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n) from outcome space $\mathcal{X} \times \mathcal{Y}$ in a time-series. Let $\mathcal{X} = \{1, \ldots, d\}$ and $\mathcal{Y} = \{0, 1\}$. By the definition of conditional probability, each distribution $P(x, y)$ over a single paired observation is the product of its marginal distribution $P(x) = \sum_y P(x, y)$ over $\mathcal{X}$ and its conditional distribution $P(y|x)$ over $\mathcal{Y}$ given x. We use this to specify two nested models called the *Conditional Single Bernoulli* model and the *Conditional Full Bernoulli* model. The distributions are extended to multiple outcomes by taking product distributions. As a consequence the pairs of observations are independently and identically distributed as seen through each model.

We define the Conditional Single Bernoulli model as

$$\mathcal{M}_2 := \{P_{0.6}\},$$

where $P_{0.6}(x) = \frac{1}{d}$ is the uniform distribution over $\mathcal{X}$ and $P_{0.6}(y|x)$ is the Bernoulli distribution with parameter 0.6, which assigns probability 0.6 to observing a one. Note that, under $P_{0.6}$, X and Y are independent. The Conditional Full Bernoulli model is defined as

$$\mathcal{M}_3 := \{P_\theta : \theta \in \Theta^d = (0, 1)^d\},$$

where $P_\theta(x) = \frac{1}{d}$ is the uniform distribution over $\mathcal{X}$ and $P_\theta(y|x)$ is the Bernoulli distribution parameterised by a separate parameter θ_x for each value of x.

Note that if $d = 1$, then the Conditional Bernoulli example reduces to the previous Bernoulli example. In our experiments, however, we will set $d = 32$, which increases the difference in complexity between the two models and magnifies the momentum problem.

3.3.2 Computing Expected Codelength

In our experiments we will sample the data from various generating sources P_{θ^*} in the Conditional Full Bernoulli model. We will summarise the codelength of individual sequences by the expected codelength under P_{θ^*}, for which we will now find efficiently computable expressions.

As the Conditional Single Bernoulli model has no parameters, it uses the same fixed distribution on all outcomes. This makes its expected codelength on n outcomes easy to compute using the sequential decomposition of expected codelength:

$$\begin{aligned} E[L_2(X^n, Y^n)] &= \sum_{i=1}^{n} E[L_2(X_i, Y_i | X^{i-1}, Y^{i-1})] \\ &= nE[L_2(X, Y)] \\ &= nE[L_2(X)] + nE[L_2(Y|X)] \\ &= n \log d + \frac{n}{d} \cdot \sum_{x} (-\theta^*_x \log 0.6 - (1 - \theta^*_x) \log 0.4). \end{aligned} \tag{3.65}$$

Also by the sequential decomposition of expected codelength, the expected codelength for the Conditional Full Bernoulli model can be rewritten as

$$\begin{aligned} E[L_3(X^n, Y^n)] &= \sum_{i=1}^{n} E[L_3(X_i, Y_i | X^{i-1}, Y^{i-1})] \\ &= \sum_{i=1}^{n} E[L_3(X_i | X^{i-1}, Y^{i-1})] + \sum_{i=1}^{n} E[L_3(Y_i | X_i, X^{i-1}, Y^{i-1})] \\ &= n \log d + \frac{1}{d} \cdot \sum_{i=1}^{n} \sum_{x_i} E[L_3(Y_i | x_i, X^{i-1}, Y^{i-1})]. \end{aligned} \tag{3.66}$$

It can be shown (See Appendix A) that the posterior probability of the next outcome according to the Conditional Full Bernoulli model using Jeffreys' Prior is given by

$$P_{\mathcal{M}_3}(Y_i = 1 | x_i, x^{i-1}, y^{i-1}) = \frac{k + \frac{1}{2}}{j + 1}, \tag{3.67}$$

where j denotes the number of occurrences of x_i in x^{i-1} and k denotes the number of occurrences of $(x_i, 1)$ in (x^{i-1}, y^{i-1}). It follows that all terms with equal j and k have the same codelength. Grouping them together gives

$$\begin{aligned} E[L_3(Y_i | x_i, X^{i-1}, Y^{i-1})] = &\sum_{j=0}^{i-1} \binom{i-1}{j} \left(\frac{1}{d}\right)^j \left(1 - \frac{1}{d}\right)^{i-1-j} \\ &\times \sum_{k=0}^{j} \binom{j}{k} {\theta^*_{x_i}}^k (1 - \theta^*_{x_i})^{j-k} \\ &\times \left(-\theta^*_{x_i} \log \frac{k + \frac{1}{2}}{j + 1} - (1 - \theta^*_{x_i}) \cdot \log \frac{j - k + \frac{1}{2}}{j + 1} \right), \end{aligned} \tag{3.68}$$

which can be computed efficiently.

3.3.3 Representative Generating Sources

We now motivate our choice of generating sources. Suppose that the expected codelength under any generating source P_{θ^*} were equal to the average of the expected codelengths under generating sources $P_{\theta^1}, \ldots, P_{\theta^d}$ with θ^i such that all parameters in θ^i are equal to the i-th component of θ^*. Then to get a good impression of the expected codelength for arbitrary generating sources in the Conditional Full Bernoulli model, it would be sufficient to obtain results for sources P_{θ^i} with all parameters equal to the same value. We will now prove this supposition, first for the Conditional Single Bernoulli model and then for the Conditional Full Bernoulli model, and conduct our experiments accordingly.

Let θ^x denote the parameter vector with all parameters equal to the x-th component of θ^*. Then $P_{\theta^*}(y|x) = P_{\theta^x}(y|x')$ for any x and x'. In addition, $L_2(x, y) = L_2(x', y)$ for any x and x'. Therefore

$$\begin{aligned}
E_{\theta^*}[L_2(X^n, Y^n)] &= n \sum_x P_{\theta^*}(x) \sum_y P_{\theta^*}(y|x) L_2(x, y) \\
&= n \sum_x P_{\theta^*}(x) \sum_{x'} P_{\theta^x}(x') \sum_y P_{\theta^*}(y|x) L_2(x, y) \\
&= n \sum_x P_{\theta^*}(x) \sum_{x'} P_{\theta^x}(x') \sum_y P_{\theta^x}(y|x') L_2(x', y) \\
&= \sum_x P_{\theta^*}(x) E_{\theta^x}[L_2(X^n, Y^n)] \\
&= \frac{1}{d} \sum_x E_{\theta^x}[L_2(X^n, Y^n)].
\end{aligned}$$

By (3.68) $E[L_3(Y_i|x_i, X^{i-1}, Y^{i-1})]$ depends on x_i only through $\theta^*_{x_i}$. In addi-

tion $L_3(x_i|x^{i-1}, y^{i-1})$ is the same for all x_i, x^{i-1} and y^{i-1}. Therefore

$$\begin{aligned}
E_{\theta^*}[L_3(X^n, Y^n)] &= \sum_{i=1}^{n} E_{\theta^*}[L_3(X_i, Y_i|X^{i-1}, Y^{i-1})] \\
&= \sum_{i=1}^{n} \sum_{x_i} P_{\theta^*}(x_i) E_{\theta^*}[L_3(x_i, Y_i|X^{i-1}, Y^{i-1})] \\
&= \sum_{i=1}^{n} \sum_{x_i} P_{\theta^*}(x_i) \sum_{x_i'} P_{\theta^{x_i}}(x_i') \\
&\qquad \times \left(E_{\theta^*}[L_3(x_i|X^{i-1}, Y^{i-1})] + E_{\theta^*}[L_3(Y_i|x_i, X^{i-1}, Y^{i-1})]\right) \\
&= \sum_{i=1}^{n} \sum_{x_i} P_{\theta^*}(x_i) \sum_{x_i'} P_{\theta^{x_i}}(x_i') \\
&\qquad \times \left(E_{\theta^x}[L_3(x_i'|X^{i-1}, Y^{i-1})] + E_{\theta^x}[L_3(Y_i|x_i', X^{i-1}, Y^{i-1})]\right) \\
&= \sum_{x} P_{\theta^*}(x) E_{\theta^x}[L_3(X^n, Y^n)] \\
&= \frac{1}{d} \sum_{x} E_{\theta^x}[L_3(X^n, Y^n)].
\end{aligned}$$

Thus the expected codelength under any P_{θ^*} is equal to the average of the expected codelengths under generating sources P_{θ^1}, ..., P_{θ^d} with θ^i such that all parameters in θ^i are equal to the i-th component of θ^*. We therefore restrict our attention to generating sources P_{θ^i} with all parameters set to the same value.

3.3.4 Results

We now show that the Switch-Point procedure gains a significant number of bits compared to MDL and Bayes due to the momentum problem. Let $L_{\mathrm{s}}(x^n)$ denote the codelength of x^n under the Switch-Point code that switches from $\mathcal{M}_2$ to $\mathcal{M}_3$. Figure 3.1 shows the expected value of $L_3(X^n) - L_2(X^n)$ and $L_{\mathrm{s}}(X^n) - L_2(X^n)$ under various P_{θ^*} for sample sizes up to 300. Each of the parameters of θ^* is set to θ_x^*. The values for θ_x^* have been chosen as representatives of qualitatively different graphs. $L_2(X^n)$ is subtracted from $L_{\mathrm{s}}(X^n)$ for ease of comparison. We will now make several observations about the results in Figure 3.1.

Consider Figures 3.2(c), 3.2(d), 3.2(e) and 3.2(g), which show the expected codelengths under P_{θ^*} with $\theta_x^* \in \{0.37, 0.38, 0.40, 0.80\}$. In all these cases the

Switch-Point code assigns the shortest codelength to the data in expectation for hundreds of sample sizes. For $\theta_x^* \in \{0.38, 0.40, 0.80\}$ the gain exceeds 3 bits for sample sizes where the expected difference in codelength between the two Conditional Bernoulli models is near zero. For $\theta_x^* = 0.80$ it even exceeds 5 bits. This gain is substantial, especially considering the small expected difference in codelength between the Conditional Bernoulli models, which never much exceeds 5 bits either. We conclude that the Switch-Point code efficiently exploits the momentum phenomenon to achieve a reduction in codelength.

In addition, the gain realised by the Switch-Point code may become even larger. Comparing the gain realised for $d = 32$ (the current example) to the gain when $d = 1$ (the original Bernoulli example) strongly suggests that for $d \gg 32$ an even larger reduction in codelength can be achieved. In addition, computational limitations prevent us from computing the Switch-Point codelength for θ_x^* even closer to 0.6. Additional plots, which are not included in this thesis, show that the expected difference in codelength between the Conditional Bernoulli models becomes much larger for such θ_x^*.

In the Bernoulli Example we observed that the momentum phenomenon occurred only if the source in the Single Bernoulli model was sufficiently close to the generating source. In Figure 3.1 we can see that for the Conditional Bernoulli Example the Switch-Point code assigns shortest codelength for many sample sizes if $\theta_x^* \in \{0.37, 0.38, 0.40, 0.80, 0.9999\}$, but not if $\theta_x^* \in \{0.20, 0.30\}$. Shortest codelength by the Switch-Point code implies that the momentum phenomenon occurs. By interpolation, we therefore assume that the momentum phenomenon occurs for many sequences if all parameters of P_{θ^*} are between 0.37 and 0.9999 and at least one parameter differs from 0.6, but not very often if all parameters are smaller than 0.30. At first sight it might appear that the region $[0.37, 0.9999]$ covers a large part of the total parameter space. However, as $d = 32$, it covers only a fraction of $(0.9999 - 0.37)^{32} = 3.77 \times 10^{-7}$, which is very small indeed.

We should be careful in drawing conclusions about typical individual sequences from the expected codelengths shown in Figure 3.1. For instance, in the figures where the Switch-Point code achieves shortest codelength the expected Switch-Point code codelength starts decreasing for sample sizes smaller than the sample size n at which the expected difference between $L_3(X^n)$ and $L_2(X^n)$ reaches its maximum. This does not happen for individual sequences, however, as the optimal switch-point for the Switch-Point code grows with the sample size until $L_3(x^n) - L_2(x^n)$ starts decreasing. The

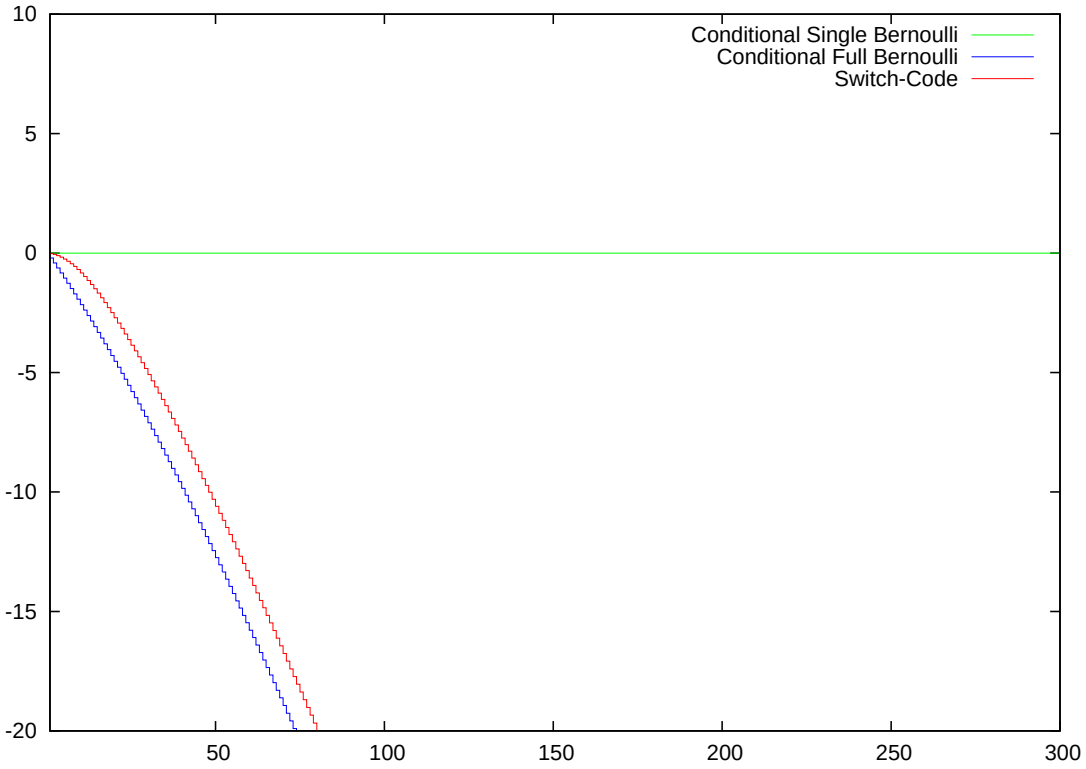
discrepancy can be explained by recalling that the expected codelength is an average over all sequences. For some of those sequences, the momentum phenomenon occurs and $L_3(x^n) - L_2(x^n)$ reaches its maximum at a sample size before its expected maximum. On those sequences the Switch-Point code gains compared to the codelengths for the Conditional Bernoulli models. Therefore the expected Switch-Point codelength starts decreasing relative to the other two expected codelengths before the expected difference in codelength between the two models reaches its maximum.

For $\theta_x^* \in \{0.37, 0.38, 0.40\}$ we note that the expected codelength of the Conditional Full Bernoulli model is smaller for small sample sizes than the expected codelength of the Conditional Single Bernoulli model. This is surprising as for these generating sources $L_2(x^n)$ is very similar to the optimal code with codelengths $-\log P_{\theta^*}(x^n)$, whereas $L_3(x^n)$ has high regret and will therefore not assign very short codelength to any short sequence. We now provide a tentative explanation of this phenomenon. Consider a typical sequence x^n sampled from generating source P_{θ^*}. Then $L_3(x^n) = -\log P_{\hat{\theta}(x^n)}(x^n) + R(P_{\mathcal{M}_3}, x^n)$ and therefore $L_3(x^n)$ is shorter than $L_2(x^n)$ if

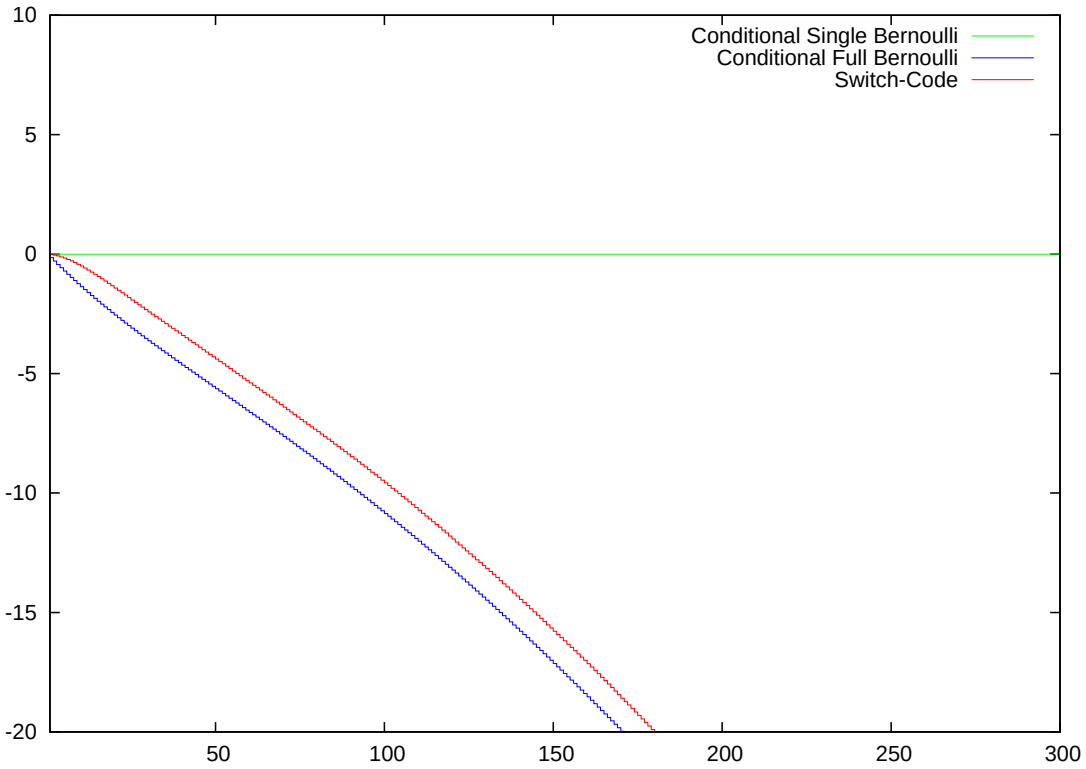
$$R(P_{\mathcal{M}_3}, x^n) < -\log P_{\theta^*}(x^n) + \log P_{\hat{\theta}(x^n)}(x^n) + \epsilon$$

for $\epsilon = L_2(x^n) + \log P_{\theta^*}(x^n)$, which is small. The difference between $-\log P_{\theta^*}$ and $-\log P_{\hat{\theta}(x^n)}(x^n)$ is very small with high probability for large sample sizes.[2] For short sample sizes, however, it is not insignificant. In particular, what we observe for $\theta_x^* \in \{0.37, 0.38, 0.40\}$ is that it exceeds $R(P_{\mathcal{M}_3}, x^n)$. This explains why the Conditional Full Bernoulli model achieves shorter codelength in expectation than the Conditional Single Bernoulli model for small sample sizes under these generating sources.

[2]Compare *typical sets* [Cover and Thomas, 1991].

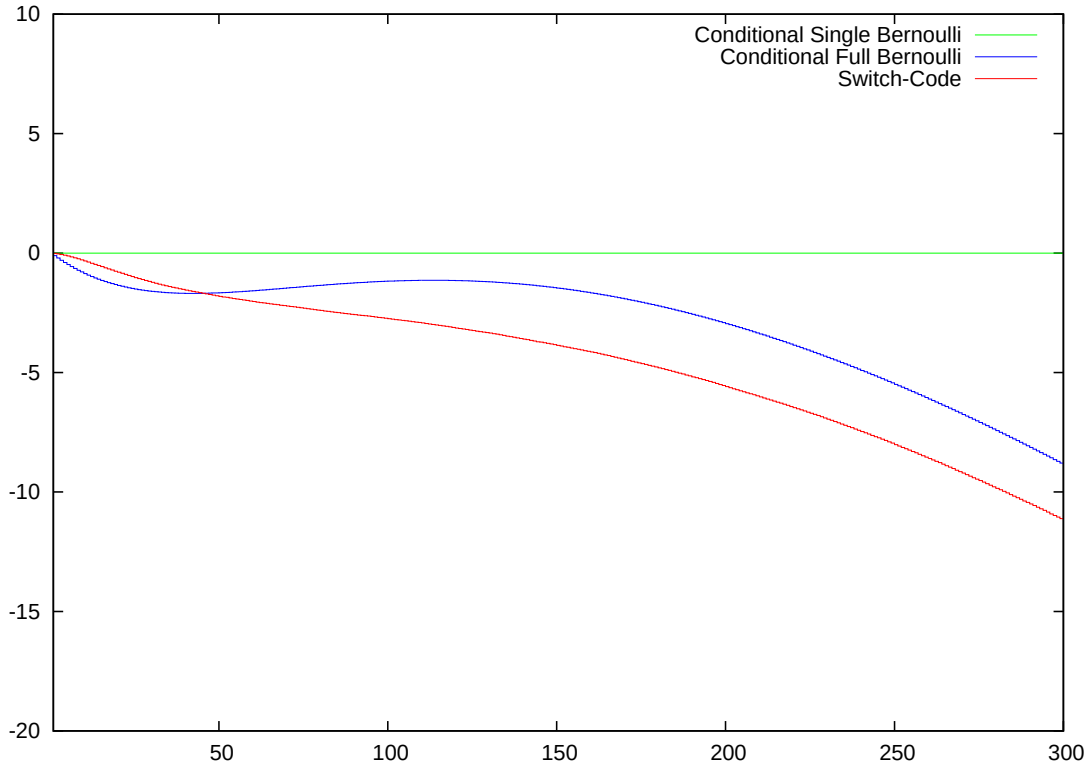


(a) $\theta_x^* = 0.20$

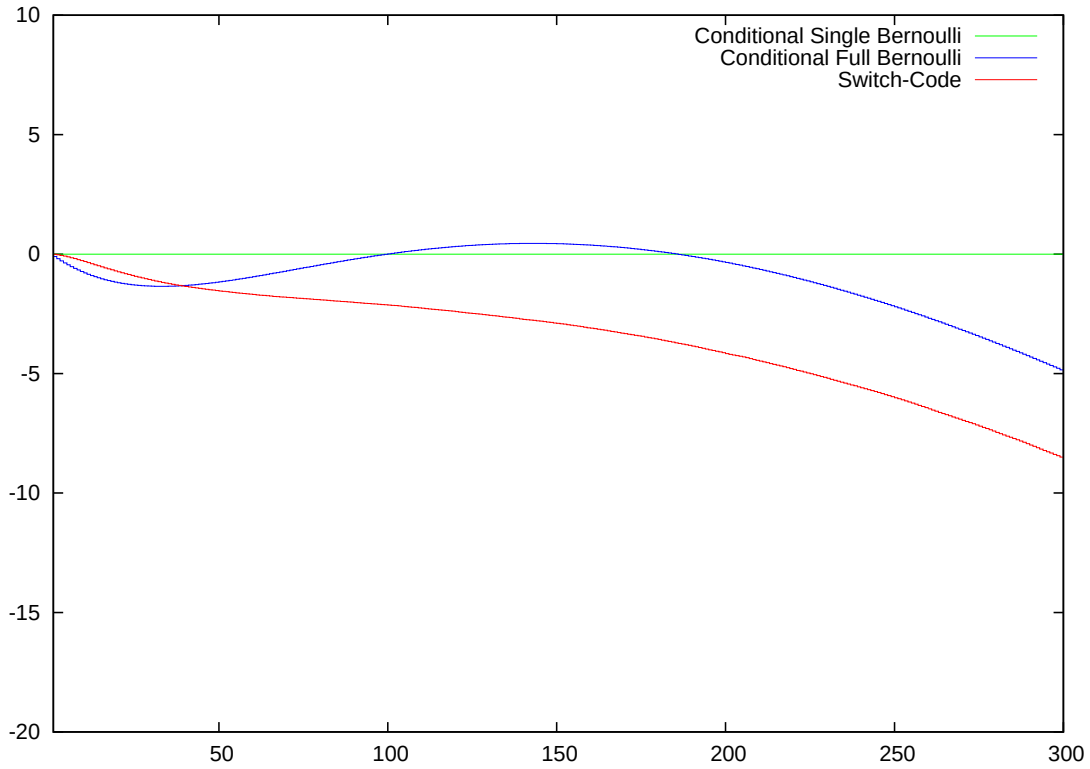


(b) $\theta_x^* = 0.30$

Figure 3.1: $E_{P_{\theta^*}}[L_3(X^n)] - E_{P_{\theta^*}}[L_2(X^n)]$ and $E_{P_{\theta^*}}[L_{\mathrm{s}}(X^n)] - E_{P_{\theta^*}}[L_2(X^n)]$ for $n = 1, \ldots, 300$. All parameters in θ^* were set to θ_x^* and $d = 32$. $E_{\theta^*}[L_{\mathrm{s}}(X^n)]$ has been approximated by averaging over 50 000 random samples from P_{θ^*}.

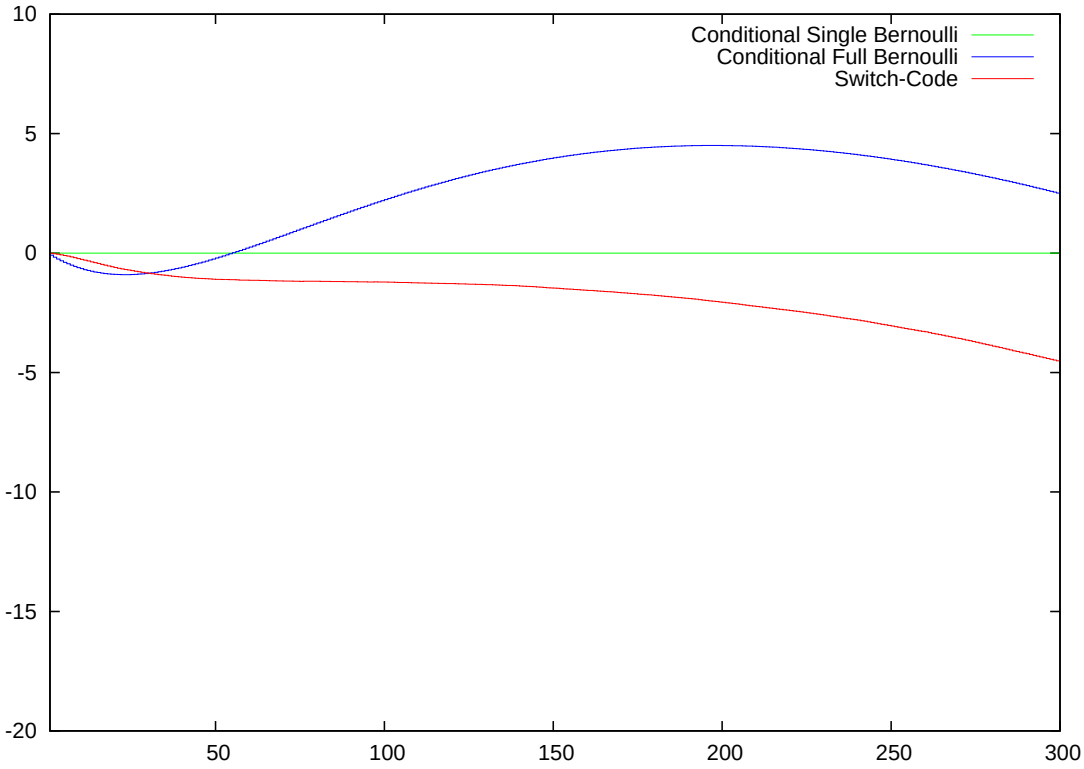


(c) $\theta_x^* = 0.37$

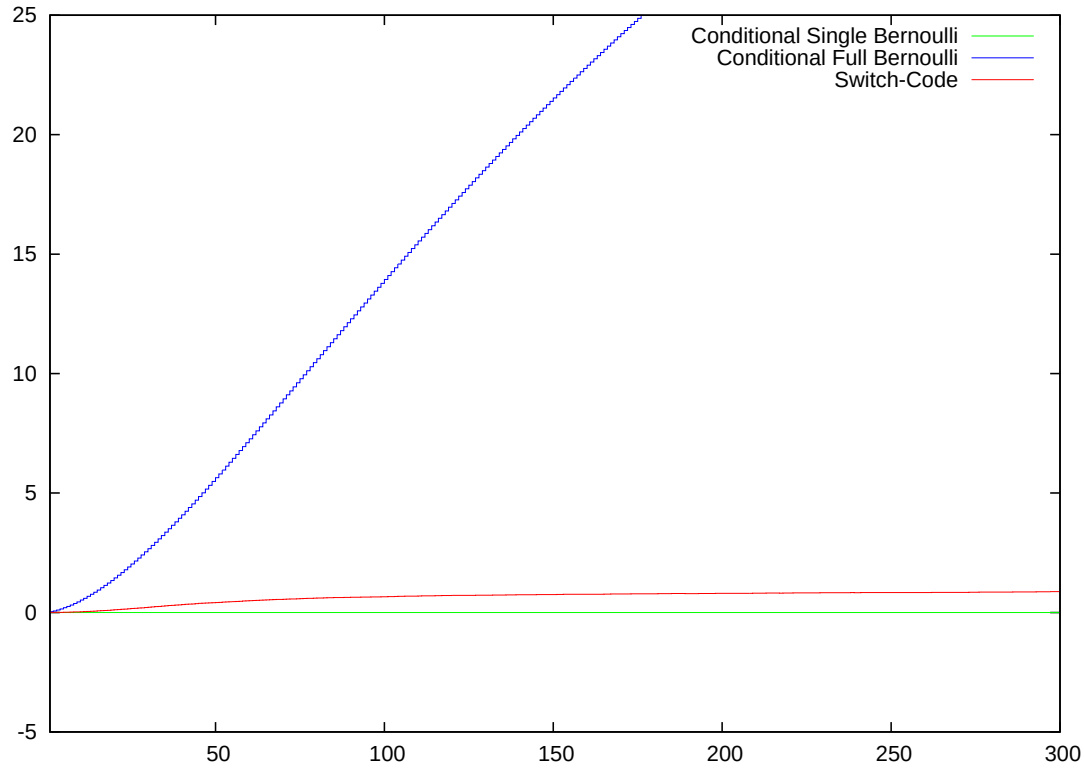


(d) $\theta_x^* = 0.38$

Figure 3.1 (cont.): $E_{P_{\theta^*}}[L_3(X^n)] - E_{P_{\theta^*}}[L_2(X^n)]$ and $E_{P_{\theta^*}}[L_s(X^n)] - E_{P_{\theta^*}}[L_2(X^n)]$ for $n = 1, \ldots, 300$. All parameters in θ^* were set to θ_x^* and $d = 32$. $E_{\theta^*}[L_s(X^n)]$ has been approximated by averaging over 50 000 random samples from P_{θ^*}.

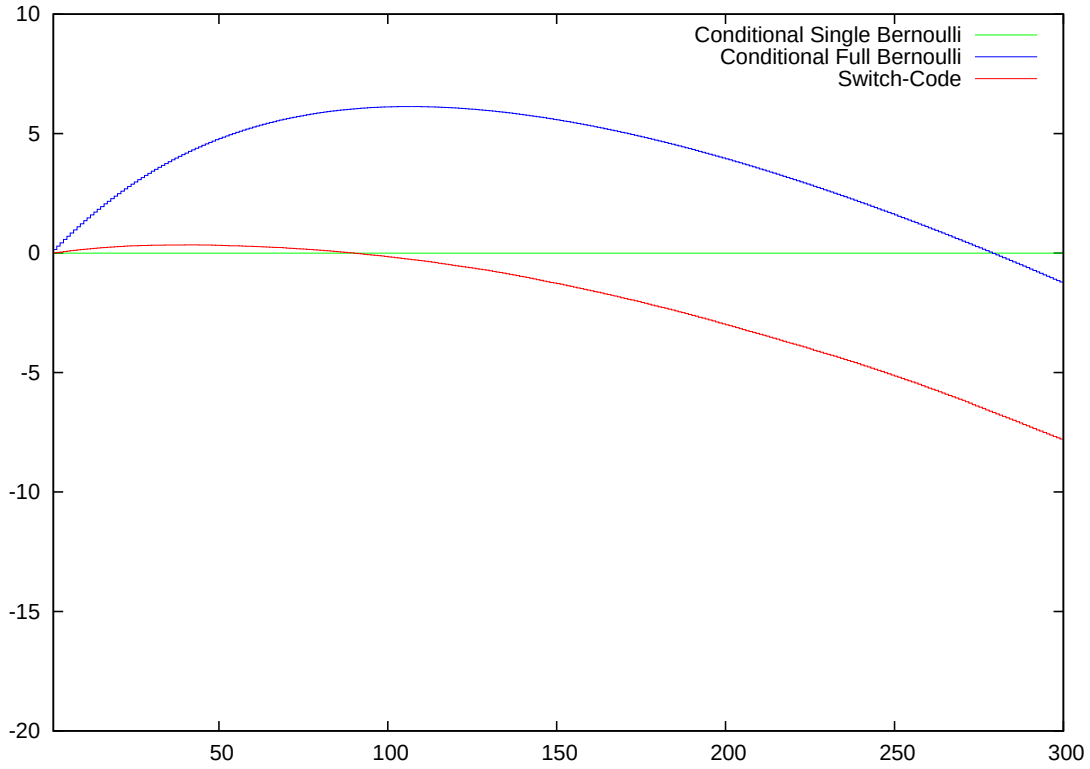


(e) $\theta_x^* = 0.40$

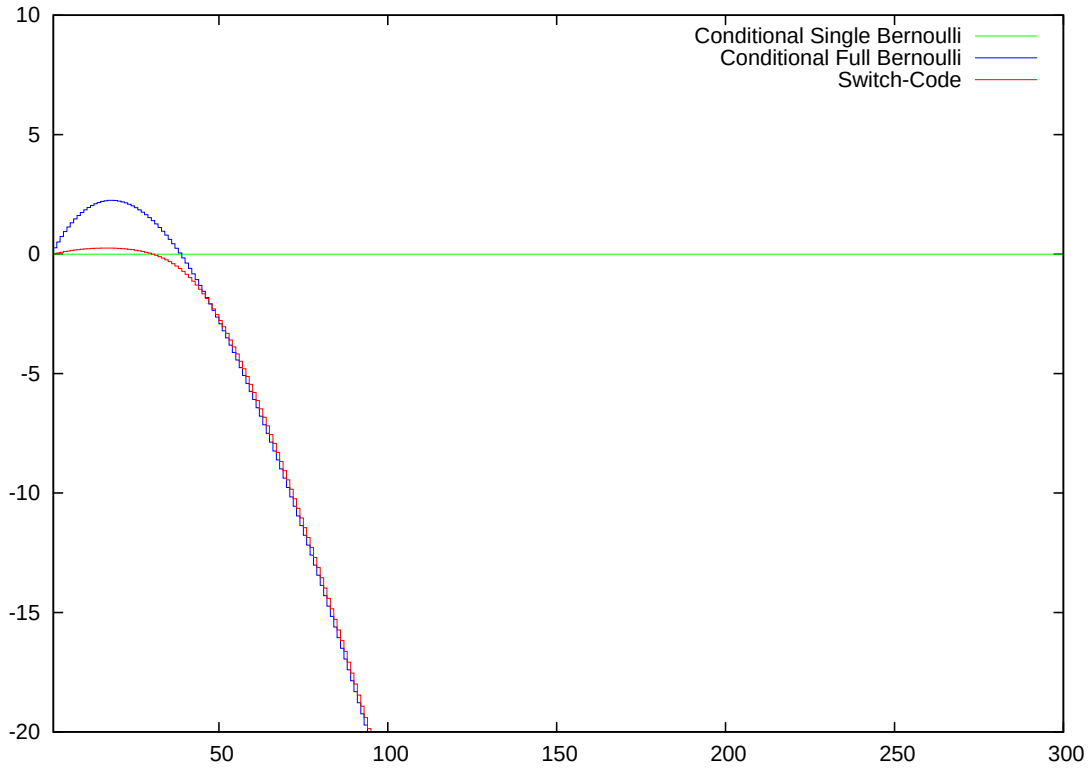


(f) $\theta_x^* = 0.60$

Figure 3.1 (cont.): $E_{P_{\theta^*}}[L_3(X^n)] - E_{P_{\theta^*}}[L_2(X^n)]$ and $E_{P_{\theta^*}}[L_s(X^n)] - E_{P_{\theta^*}}[L_2(X^n)]$ for $n = 1, \ldots, 300$. All parameters in θ^* were set to θ_x^* and $d = 32$. $E_{\theta^*}[L_s(X^n)]$ has been approximated by averaging over 50 000 random samples from P_{θ^*}.



(g) $\theta_x^* = 0.80$



(h) $\theta_x^* = 0.9999$

Figure 3.1 (cont.): $E_{P_{\theta^*}}[L_3(X^n)] - E_{P_{\theta^*}}[L_2(X^n)]$ and $E_{P_{\theta^*}}[L_s(X^n)] - E_{P_{\theta^*}}[L_2(X^n)]$ for $n = 1, \ldots, 300$. All parameters in θ^* were set to θ_x^* and $d = 32$. $E_{\theta^*}[L_s(X^n)]$ has been approximated by averaging over 50 000 random samples from P_{θ^*}.

3.4 Chapter Summary

We have proved that in the Bernoulli example the momentum phenomenon can get arbitrarily large with arbitrarily high probability for any generating source in a suitably chosen subset Ψ of the Full Bernoulli model. We argued that the minimum number of outcomes at which regular MDL and Bayes predict suboptimally, increases with the size of the momentum phenomenon. It follows that we can get the number of outcomes at which MDL and Bayes predict suboptimally arbitrarily large with arbitrarily high probability for generating sources in a suitably chosen Ψ. In addition, we proved that the momentum phenomenon could get arbitrarily large in expectation for all generating sources in a suitable Ψ on any two-model prediction problem if one of the models was an exponential family with positive dimension and the other model was any submodel containing only a single probabilistic source.

The momentum phenomenon was small in the Bernoulli example. We therefore generalised it to the Conditional Bernoulli example, which provides a concrete example where the Switch-Point procedure gains a significant number of bits compared to regular MDL and Bayes. It increased the difference in rates of convergence between the models by increasing the difference in complexity. We demonstrated for the Conditional Bernoulli example that for some generating sources the Switch-Point code gained 3 or even 5 bits in expectation compared to the best model. This was comparable to the maximum expected difference in codelength between the original models. As a result the Switch-Point procedure reduced predictive loss compared to regular MDL and Bayes in these cases. Furthermore, we argued that if either the difference in complexity between the Conditional Bernoulli model were increased by taking $d \gg 32$, or the generating source were taken even closer to the source in the Single Bernoulli model by setting θ_x^* closer to 0.6, then the Switch-Point code should gain even more bits. Finally, just as in the Bernoulli example, we observed that the momentum phenomenon occurred if the source in the Conditional Single Bernoulli was sufficiently similar to the generating source.

CHAPTER 4

Discussion

This chapter provides a discussion of the momentum problem. We first offer an explanation in terms of strategies for prediction, which are constructed in two stages. Then we provide a discussion of the Switch-Point procedure and consider the implications of our results for the task of model selection. Finally, we review related prior work and make several suggestions for future research.

4.1 Understanding the Momentum Problem

In this section we give an interpretation of the momentum problem in terms of strategies for prediction, which are called prequential forecasting systems (PFSs). These are the subject of interest in the so-called prequential approach to statistics. We interpret MDL (and Bayesian Model Averaging) as the construction of PFSs in two stages. We then argue that the design of regular MDL does not take the momentum into account in the second stage. This explains why the Switch-Point procedure, which has been explicitly designed with the momentum phenomenon in mind, is able to improve predictive accuracy compared to regular MDL

4.1.1 Prequential Approach to Statistics

We introduced the Bernoulli example by considering *daily* forecasts of the probability of precipitation for the next day. Likewise we will now consider predicting the next outcome in a time-series at *every* moment in the series. That is, we consider not only $P(x_{n+1}|x^n)$, but the entire sequence of predictions $P(x_1)$, $P(x_2|x^1)$, ..., $P(x_{n+1}|x^n)$. This hits upon the *prequential approach* to statistical theory, which is based on the premise that such sequential probability forecasting is the purpose of statistical inference [Dawid, 1984].

The prequential approach is based on *prequential forecasting systems* (PFSs). A PFS is defined as a rule that specifies a conditional distribution on outcome x_{n+1} given any possible sequence of n observations of the past at *every* sample size n. Any PFS defines a unique sequence of conditional probability distributions and vice versa. As shown in Section 1.2.1, any sequence of conditional distributions defines a unique probabilistic source and vice versa. It follows that PFSs and probabilistic sources are *mathematically* equivalent [Dawid, 1992b]. Nevertheless, they are usually given different *interpretations*. A probabilistic source is commonly viewed as a potential *explanation* for the data while a PFS is seen as a *strategy* for predicting the data.

4.1.2 Prediction using PFSs

Previously, we have described MDL and Bayes in terms of probabilistic sources. Without changing the two methods we now substitute PFSs instead. Thus, we view each probabilistic source in a model as a PFS. In addition, the sequence of universal models $P^1_{\mathcal{M}}$, $P^2_{\mathcal{M}}$, $\cdots$ for each model $\mathcal{M}$ defines the PFS $P_{\mathcal{M}}(x_1)$, $P^2_{\mathcal{M}}(x_2|x^1)$, $P^2_{\mathcal{M}}(x_3|x^2)$, $\cdots$. We let both sequences be denoted by $P_{\mathcal{M}}$ since they are mathematically equivalent and only differ in their interpretations. Likewise, from now on we interpret P_{MDL} (or P_{Bayes}) as a PFS, which combines the PFSs $P_{\mathcal{M}}$ for the models (see (1.16) or (1.12)). The interpretation of $P_{\mathcal{M}}$ as a PFS is standard [Dawid, 1992b; Grünwald et al., 2005]. The interpretation of P_{MDL} (or P_{Bayes}) as a PFS, on the other hand, is less common.

The task of prediction in the presence of multiple models has thus been decomposed into two (interdependent) stages (see Figure 4.1). Stage one is to construct a PFS $P_{\mathcal{M}}$ for each model $\mathcal{M}$ separately. Stage two is to

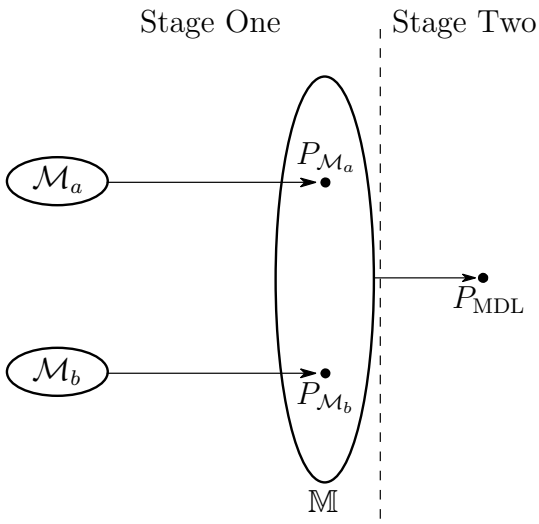


Figure 4.1: Construction of PFSs in two stages in regular MDL.

construct a single PFS P_{MDL} out of the individual PFSs $P_{\mathcal{M}}$. The notion of a PFS has been applied on three levels: first for individual predictors in the models, then to construct a predictor for each model in stage one, and finally to construct a single strategy for prediction based on the separate predictors for the models in stage two. It should be stressed that assumptions and requirements about the PFSs may differ between levels. We will return to this in the next section.

Stage One

In stage one alternative approaches for constructing the PFS $P_{\mathcal{M}}$ for each model $\mathcal{M}$ are available. For instance, we might use the maximum likelihood PFS in $\mathcal{M}$:

$$P_{\mathcal{M}}(x_{n+1}|x^n) = P_{\hat{\theta}(x^n)}(x^{n+1}),$$

where

$$P_{\hat{\theta}(x^n)} := \arg\max_{P_\theta \in \mathcal{M}} P_\theta(x^n) = \arg\min_{P_\theta \in \mathcal{M}} \sum_{i=1}^{n} -\log P_\theta(x_i|x^{i-1})$$

is the best predictor of all past observations as measured by accumulated predictive loss. This approach is called the *predictive* MDL principle [Rissanen, 1986a]. If the data are sampled from a source in $\mathcal{M}$, then, under regularity

conditions on $\mathcal{M}$, it can be shown that $P_{\hat{\theta}(x^n)}$ acts as an efficient universal model relative to $\mathcal{M}$ with probability one [Grünwald et al., 2005], but see [Grünwald and de Rooij, 2005] for caveats.

Stage Two

As in stage one, alternatives are possible for constructing P_{MDL} in stage two. For simplicity, we consider the situation with two models in the model set. In this case such an alternative is provided by the Switch-Point procedure. It adds the Switch-Point model $\mathcal{M}_{\mathrm{s}}$ (see Figure 4.2), which before has been defined as the set of sources that code the first s outcomes using the source for model $\mathcal{M}_a$ and the rest of the outcomes using the source for model $\mathcal{M}_b$, with switch-point s ranging over the nonnegative integers. In terms of probabilistic sources the Switch-Point model may be considered a strange combinator of the probabilistic sources for the models. Viewed as a set of PFSs, however, it expresses a natural strategy for prediction that anticipates the momentum phenomenon. Each PFS in $\mathcal{M}_{\mathrm{s}}$ follows the predictions of the PFS $P_{\mathcal{M}_a}$ for $\mathcal{M}_a$ up to its switch-point and then uses the predictions of the PFS $P_{\mathcal{M}_b}$ for $\mathcal{M}_b$ to predict subsequent outcomes.

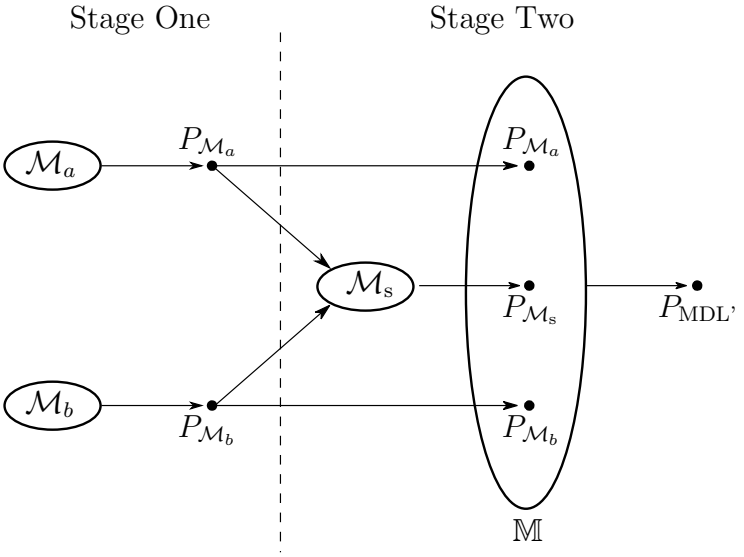


Figure 4.2: Construction of PFSs in two stages in the Switch-Point procedure.

The PFS $P_{\mathcal{M}_{\mathrm{s}}}$ for $\mathcal{M}_{\mathrm{s}}$ that has been defined in (2.1), is called the Switch-Point code. It predicts according to

$$\begin{aligned} P_{\mathcal{M}_{\mathrm{s}}}(x_{n+1}|x^n) &= \frac{P_{\mathcal{M}_{\mathrm{s}}}(x^{n+1})}{P_{\mathcal{M}_{\mathrm{s}}}(x^n)} \\ &= \sum_{s=0}^{\infty} P_{\mathcal{M}_{\mathrm{s}}}(x_{n+1}|x^n, s) \cdot w_s(s|x^n) \\ &= P_{\mathcal{M}_b}(x_{n+1}|x^n) \sum_{s=0}^{n} w_s(s|x^n) + P_{\mathcal{M}_a}(x_{n+1}|x^n) \left(1 - \sum_{s=0}^{n} w_s(s|x^n)\right). \end{aligned}$$

This is a mixture of the predictions according to $P_{\mathcal{M}_a}$ and $P_{\mathcal{M}_b}$. The ratio in the mixture depends on the posterior distribution over the switch-points $w_s(s|x^n)$ given observations x^n. In particular, it depends on the probability according to $w_s(s|x^n)$ of whether the switch-point has already been passed.

As an aside, we point out that $P_{\mathcal{M}_{\mathrm{s}}}$ can efficiently be computed using the fact that

$$P_{\mathcal{M}_{\mathrm{s}}}(x^n) = P_{\mathcal{M}_b}(x^n) \sum_{s=0}^{n} w_s(s) \frac{P_{\mathcal{M}_a}(x^s)}{P_{\mathcal{M}_b}(x^s)} + P_{\mathcal{M}_a}(x^n) \left(1 - \sum_{s=0}^{n} w_s(s)\right).$$

In addition, we note that we have slightly extended Dawid's use of the term prequential forecasting system. Dawid only applies it to the PFSs constructed in stage one and, more recently, also to the PFSs constructed in stage two [Dawid, 1992a, p. 117]. He does not link the individual probabilistic sources in each model to PFSs.

4.1.3 Different Assumptions between Stages

We have said that assumptions and requirements about the PFSs may differ between stages one and two. This insight is crucial in understanding the momentum problem. We will now elaborate and argue that the momentum problem is the result of assumptions implicit in the regular MDL and Bayesian methods that are justified in stage one, but not in stage two.

Stage One

In stage one, Bayesian universal models with Jeffreys' prior are used to approximate the optimal NML universal model (see Section 1.4). Recall that the NML universal model for a model $\mathcal{M}$ minimises worst-case regret. That is, it minimises the worst-case additional codelength compared to the best PFS $P_{\hat{\theta}(x^n)}$ in $\mathcal{M}$ for the data x^n. In general, a Bayesian universal model based on prior w over the elements of $\mathcal{M}$ achieves (reasonably) small regret on x^n if $w(\hat{\theta}(x^n))$ is (reasonably) large (see also Section 1.2.7). To construct $P_{\mathcal{M}}$ as a Bayesian universal model for model $\mathcal{M}$ therefore means to assign nearly as short codelength to the data as the best PFS in $\mathcal{M}$ for that data. For the individual PFSs in $\mathcal{M}$ it may — by the interpretation of a model — be assumed that at least one of them describes the data well. Therefore $P_{\mathcal{M}}$ assigns nearly as short codelength to the data as a PFS that describes the data well. This justifies constructing $P_{\mathcal{M}}$ as a Bayesian universal model at each sample size.

Stage Two

In stage two, the regular MDL procedure constructs P_{MDL} as a Bayesian universal model for the model set. Just like in stage one this choice is motivated by examining its (worst-case) regret. The regret of P_{MDL} is defined as the additional codelength that it assigns to the data, x^n, compared to the code $P_{\mathcal{M}}$ that minimises the codelength for x^n among all models in the model set. When there are only few models, as in this thesis, then the regret of P_{MDL}, which is based on a uniform prior $w_{\mathbb{M}}$ over the model set, is always small (see page 15). It follows that the worst-case regret of P_{MDL} must also be small.

The (worst-case) regret is a good indicator of predictive accuracy if and only if the code $P_{\mathcal{M}}$ for at least one model $\mathcal{M}$ describes the data well at all sample sizes. However, in stage two there is no reason to assume that such a single optimal model should always exist. In fact, it is this assumption that is violated whenever the momentum phenomenon occurs! Viewed differently, if we think of the choice of model as a parameter that needs to be estimated, then the momentum phenomenon implies that the value of this parameter is non-constant in the sample size. We conclude that (worst-case) regret is not the appropriate measure to minimise when the momentum phenomenon occurs. The design of regular MDL, therefore, does not sufficiently take the momentum phenomenon into account.

This strongly suggests that it should be possible to improve on regular MDL whenever the momentum phenomenon occurs, since we have shown in Sections 3.1 and 3.2 that the momentum phenomenon may get very large. Moreover, in Section 3.3 we have shown that the Switch-Point procedure, which refines regular MDL by explicitly modelling at which sample size the best-predicting model switches, can sometimes actually realise the suggested improvement. In addition, we will show in Section 4.2 that the Switch-Point procedure can never assign much longer codelength to the data than regular MDL. We conclude that regular MDL may behave suboptimally when faced with the momentum *phenomenon*. Therefore the momentum phenomenon should, in fact, be considered a momentum *problem*. This is the most important insight of this thesis.

4.2 Discussion of the Switch-Point Procedure

In Sections 2.3 and 3.3 we found that the Switch-Point procedure may improve predictive performance compared to regular MDL. We will now discuss some additional characteristics of the Switch-Point procedure.

4.2.1 Applicability

We claimed in Section 2.2 that the worst-case increase in codelength due to adding the Switch-Point model compared to regular MDL is very small relative to the total codelength for the data. Formally, this can be verified by

$$\begin{aligned} -\log \sum_{\mathcal{M}\in\mathbb{M}\cup\{\mathcal{M}_\mathrm{s}\}} \frac{1}{3}\cdot P_{\mathcal{M}}(x^n) - P_{\mathrm{MDL}}(x^n) &= -\log \frac{\sum_{\mathcal{M}\in\mathbb{M}\cup\{\mathcal{M}_\mathrm{s}\}} \frac{1}{3}\cdot P_{\mathcal{M}}(x^n)}{\sum_{\mathcal{M}\in\mathbb{M}} \frac{1}{2}\cdot P_{\mathcal{M}}(x^n)} \\ &\leq -\log \frac{2}{3} \\ &\approx 0.58, \end{aligned}$$

where the worst case is achieved only if $P_{\mathcal{M}_\mathrm{s}}(x^n) = 0$. In fact, this worst case will most likely never be achieved as it follows directly from (2.1) that for any data sequence x^n

$$L_\mathrm{s}(x^n) \geq -\log w_s(0) P_{\mathcal{M}_b}(x^n),$$

and hence $P_{\mathcal{M}_s}(x^n) \geq w_s(0)P_{\mathcal{M}_b}(x^n)$, which will typically be greater than zero. There is therefore little risk in using the Switch-Point procedure instead of regular MDL (or Bayesian Model Averaging). As seen in Section 3.3, however, the potential gains are significant. A modification to a code that satisfies these two properties is called an application of the *luckiness principle* in [Grünwald, 2007], because we can gain significantly if we are lucky, but will hardly lose anything if we are not. It is therefore always a safe bet to use the Switch-Point procedure.

The addition of the Switch-Point model decreases total codelength if the Switch-Point code assigns shorter codelength to the data than the codes for the other models. Suppose the momentum phenomenon occurs with size C. Then the code that switches at the optimal switch-point $\hat{s}$ will achieve C bits shorter codelength than the codes for the models $\mathcal{M}_a$ and $\mathcal{M}_b$. The Switch-Point code exceeds this by at most $-\log w_s(\hat{s})$ bits. Therefore

$$-\log P_{\mathcal{M}_s}(x^n) \leq \min_{\mathcal{M}\in\mathbb{M}} -\log P_{\mathcal{M}}(x^n) - C + [-\log w_s(\hat{s})]. \tag{4.1}$$

In other words, the Switch-Point code gains at least $C - [-\log w_s(\hat{s})]$ bits if the momentum phenomenon occurs. We note that this gain does not depend on the total sample size n.

The bound in (4.1) ignores the probability assigned to the data by all probabilistic sources in the Switch-Point model with switch-points s that are near, but not exactly at, $\hat{s}$. These sources assign nearly as much probability to the data as the source that switches exactly at $\hat{s}$. Hence we have ignored a significant part of the probability assigned to the data by $P_{\mathcal{M}_s}$. Presumably, (4.1) is therefore not a tight bound on the gain that is realised by the Switch-Point code.

Still, we expect the gain of the Switch-Point code to decrease with increasing $\hat{s}$ as then $w_s(s)$ will typically be small for all s near $\hat{s}$. This is the case in the proofs from Sections 3.1 and 3.2. There $\hat{s}$ — called n_1 — got very large. It is therefore not clear whether the Switch-Point code is always able to achieve significant reductions in codelength when the momentum phenomenon gets very large. However, preliminary work on a refinement of the Switch-Point code, which there was no time to report on here, suggests that it is possible to gain at least $2^{\lfloor \log C \rfloor} - \lfloor \log C \rfloor$ bits. In this refinement the gain depends only on the size of the momentum phenomenon and not on the switch-point.

The Switch-Point procedure will only decrease codelength if the momentum phenomenon occurs. The momentum phenomenon can occur if there are

models in the model set that require a different number of observations to learn good values for their parameters. In the experiments in this thesis this was achieved by selecting models of different complexity. However, preliminary additional experiments, not reported on in this thesis, suggest that it is possible to construct model selection scenarios in which models that are identical up to a symmetric transformation and therefore have the same complexity, still require a different number of observations to learn their parameters[1]. This shows the difficulty in completely ruling out the possibility that the momentum phenomenon might occur. It may therefore be prudent to apply the Switch-Point procedure instead of regular MDL or BMA in many cases, even when the models are not nested.

4.2.2 Models as Black Boxes

We might consider whether the PFSs $P_{\mathcal{M}}$ for the models (see Figure 4.1) should perhaps always be constructed such that the momentum phenomenon is avoided. Suppose, however, that we view the PFS for each model as a formalisation of the predictive strategy of an expert. This expert might, for instance, be a meteorologist who predicts the probability of precipitation just like in the introduction of the Bernoulli example in Section 2.1. In this case it would seem inappropriate to demand that the experts coordinate their predictions in order to avoid the momentum phenomenon. The Switch-Point procedure therefore makes no assumptions about the construction of PFSs for the individual models. It considers these PFSs *black boxes* that generate predictions. They may be constructed from the models using subjective Bayesian priors based on prior belief, objective Bayesian priors based on practical considerations, using the maximum likelihood PFS or in any reasonable way at all.

Approaching the predictive distributions for the models as black box procedures makes the Switch-Point code compatible with approaches to combining predictions by multiple experts as considered by Bousquet and Warmuth [2002]. Connections to their work are discussed further in Section 4.7, which reviews options for future work.

[1] In these experiments the generating distribution was not contained in any of the models.

4.2.3 The Switch-Point Procedure Builds on Existing Methods

The Switch-Point procedure can be interpreted as regular MDL prediction with the Switch-Point model added to deal with the momentum phenomenon. As such it can easily be compared to those existing methods and many theoretical results [e.g. Barron et al., 1998] transfer with little modification. In addition the Switch-Point procedure inherits all the strengths of MDL.

4.2.4 Alternative Switch-Point Model with Fixed-Ratio Mixtures

We will now substantiate our claim from Section 2.2 that the Switch-Point model $\mathcal{M}_{\mathrm{s}}$ should not be constructed as a set of fixed-ratio mixtures of $P_{\mathcal{M}_a}$ and $P_{\mathcal{M}_b}$. That is, we consider defining $\mathcal{M}_{\mathrm{s}}$ as

$$\mathcal{M}_{\mathrm{s}} := \{\theta P_{\mathcal{M}_a}(x^n) + (1-\theta)P_{\mathcal{M}_b}(x^n) : \theta \in [0,1]\},$$

which might perhaps be proposed by a Bayesian who would consider putting some prior $w(\theta)$ on the relative weight, θ, of $P_{\mathcal{M}_a}$ compared to $P_{\mathcal{M}_b}$. However, for any $\theta \in [0,1]$ we have that

$$\theta P_{\mathcal{M}_a}(x^n) + (1-\theta)P_{\mathcal{M}_b}(x^n) \leq \max(P_{\mathcal{M}_a}(x^n), P_{\mathcal{M}_b}(x^n)), \tag{4.2}$$

which does not achieve the goal to construct the elements of the Switch-Point model such that they assign short codelength to the data if the best predicting model changes from $\mathcal{M}_a$ to $\mathcal{M}_b$. By contrast, if the Switch-Point model is defined as in Section 2.2, then the PFS in the Switch-Point model that switches at the optimal switch-point will gain a number of bits equal to the size of the momentum phenomenon compared to both $P_{\mathcal{M}_a}$ and $P_{\mathcal{M}_b}$.

4.2.5 Many Models

We now consider extending the Switch-Point procedure to prediction when there are more than two models. Suppose that in such a setting two models could be identified that could be anticipated to exhibit the momentum phenomenon. Then a Switch-Point model might be added that switched between these two models.

Alternatively, it is conceivable that in prediction with three or more models $\mathcal{M}_a$, $\mathcal{M}_b$, $\mathcal{M}_c$, etc., the best predicting model could be anticipated to change two times: first from $\mathcal{M}_a$ to $\mathcal{M}_b$ and then from $\mathcal{M}_b$ to $\mathcal{M}_c$. In such a scenario a model $\mathcal{M}_\mathrm{s}$ might be added with two switch-points: s_1 to switch from $\mathcal{M}_a$ to $\mathcal{M}_b$ and s_2 to switch from $\mathcal{M}_b$ to $\mathcal{M}_c$. Compared to the two-model prediction problem on only models $\mathcal{M}_a$ and $\mathcal{M}_c$ with the Switch-Point code switching directly from $\mathcal{M}_a$ to $\mathcal{M}_c$ at switch-point s, the optimal values for both s_1 and s_2 will tend to be different from the optimal value for s. It can be concluded that the addition of model $\mathcal{M}_b$ changes the relationship between model $\mathcal{M}_a$ and model $\mathcal{M}_c$ under this extension of the Switch-Point procedure to prediction with more than two models.

4.3 Model Selection

Though our results were obtained for prediction, they have implications for the task of model selection as well. Recall from Section 1.4 that in model selection MDL codes the data using a two-part code, which first selects a model and then codes the data with the help of that model. The Switch-Point model represents the hypothesis that the best-predicting model will change in the sample size. If the corresponding Switch-Point code achieves shorter codelength than the codes for the original models, then we might wonder whether its hypothesis may actually be a better explanation for the data than either of the original models.

If the Switch-Point model were added to the model set, then the corresponding two-part code would select it whenever it achieved shortest codelength among all models. It is not obvious, however, whether we would be justified in adding it. This can be verified using the MDL principle, which dictates that the addition of the Switch-Point model is an improvement if the resulting two-part code can be expected to assign shorter codelength to the data than the original two-part code that did not consider the Switch-Point model. We can see when this is the case in our experiments, which will be reconsidered below from the perspective of model selection. It will be seen that in some cases the addition of the Switch-Point model is justified. From this we conclude that the momentum problem transfers to model selection as well. This is the third insight of this thesis.

Assuming a uniform prior $w_\mathcal{P}$ over the models, the addition of the Switch-Point model increases the first part of the two-part code by $\log 3 - \log 2 \approx 0.58$

bits. Adding it is therefore justified if it can be anticipated that it will make up for this overhead in the second part of the code. That is, adding the Switch-Point model is justified if the Switch-Point code is expected to achieve at least 0.58 bits shorter codelength than the best of the original models on typical sequences. There will also exist sequences on which the Switch-Point code does not reduce total codelength. In the worst case, which is achieved if the Switch-Point model is not selected, the addition of the Switch-Point model increases codelength by 0.58 bits. This is small, so adding it is never a big risk. We will now reexamine our results to see when the addition of the Switch-Point code reduces total codelength.

First we consider the Bernoulli example. In Figure 2.3 we compared the Switch-Point code codelength to the codelengths of the codes for original models on individual sequences that were sampled from different generating distributions. In Figures 2.3(a), 2.4(e), 2.4(f) and 2.4(g), which correspond to the sequences sampled with θ^* equal to 0.6, 0.7, 0.3 and 0.9 respectively, the Switch-Point code clearly does not achieve the required reduction in codelength. Figures 2.4(b), 2.4(c) and 2.4(d), which respectively correspond to θ^* equalling 0.55, 0.65 and 0.5, are not so easy to read. Inspection of the raw data, however, reveals that for $n \geq 800$ the Switch-Point code gains approximately 0.47 bits for $\theta^* = 0.55$ and 0.69 bits for $\theta^* = 0.65$ compared to the best of the other two codes. For $\theta^* = 0.5$ it gains at least 0.9 bits for all sample sizes. The addition of the Switch-Point model therefore reduces total codelength for the sequences sampled with θ^* equal to 0.65 and 0.5.

In Figure 2.4 we compared expected codelengths in the Bernoulli example. As on the individual sequences, the Switch-Point code does not reduce total expected codelength under θ^* equal to 0.6, 0.3 and 0.9, which are shown in Figures 2.4(a), 2.5(f) and 2.5(g) respectively. Under θ^* equal to 0.5 and 0.7, shown in Figures 2.5(d) and 2.5(e) respectively, it reduces total expected codelength for about 20 sample sizes around 90 and 100 respectively. Finally, Figures 2.5(b) and 2.5(c), which show expected codelength under θ^* equalling 0.55 and 0.65 respectively, show that the Switch-Point code achieves approximately 1.5 bits shorter codelength for sample sizes over 560. In these cases the addition of the Switch-Point model therefore reduces total codelength by approximately 0.92 bits, which is small, but may be considered significant considering that the momentum phenomenon is also small in the Bernoulli example. We conclude that based on our experiments it remains unclear whether the addition of the Switch-Point model is justified in the Bernoulli example.

We now consider the Conditional Bernoulli example. In Figure 3.1 we compared the expected Switch-Point code codelength to the expected codelengths for the two Conditional Bernoulli models. In Figures 3.1(a), 3.1(b), 3.2(f) and 3.2(h), which show the expected codelengths under θ_x^* equal to 0.2, 0.3, 0.6 and 0.9999 respectively, the Switch-Point code does not achieve the required reduction in expected codelength. However, in Figures 3.2(c), 3.2(d), 3.2(e) and 3.2(g), which correspond to θ_x^* equal to 0.37, 0.38, 0.4 and 0.8 respectively, it does reduce total expected codelength. In these cases the number of bits gained by adding the Switch-Point model ranges from approximately 2 to 5 bits. As argued in Section 3.3, the gain may be even larger if the difference in complexity between the models were scaled up by taking $d \gg 32$ or by taking θ_x^* even closer to 0.6. We conclude that the addition of the Switch-Point model may significantly reduce total codelength and is therefore justified in the Conditional Bernoulli example.

4.4 Prior Work

Though the momentum problem has not been extensively investigated in any prior work, several existing results suggest that it has been encountered before. This section sheds more light on some of those results by explaining them in terms of the momentum phenomenon and the momentum problem. All prior work that we will consider, is on model selection. We first recognise the momentum phenomenon in a discussion among experts on Dawid's prequential approach to statistics. Then, similarities between MDL and the so-called predictive least squares principle suggest an occurrence of the momentum problem in regression among normal linear models. Furthermore, we challenge the claimed predictive optimality of the so-called median probability model. And finally, we briefly mention the method of forward validation, which may be related to our work.

4.4.1 Start-up Problem

In [Dawid, 1992b] the prequential approach to statistics is connected to MDL and Bayes factors model selection. This paper is followed by a discussion among experts. In this discussion, J. Rissanen writes:

> "[A]ttention must be paid to an inherent start-up problem, which arises from poor initial predictions. Unless checked, these can grossly penalize a complex model, which at the later stages may turn out to provide superior predictions."

We may interpret Rissanen's words in two ways. The first, suggested by his other writings [e.g. Rissanen, 1989, 1986a], is that Rissanen is thinking of an entirely different problem that arises in predictive MDL (see Section 4.1.2) when the maximum likelihood estimator assigns probability zero — and therefore infinite codelength — to the next outcome. The other possible interpretation is that Rissanen is warning us about something like the momentum problem. This latter interpretation fits best with the response by A. P. Dawid, who writes:

> "It seems to me perfectly reasonable that a complex model should be heavily penalized in the initial stages, since the slow rate at which its parameters can be learned means that it may for a long time predict more poorly than a simpler 'incorrect' model. In this case I would rather use the simple model until the data are sufficiently extensive as to demand more detailed description. In general, the complexity of the model used should increase with the amount of data available[.]"

Following Dawid, we therefore adopt the second interpretation. In this case it is exactly such a start-up problem that we have captured by our definition of the momentum phenomenon. Using insights gained from the momentum phenomenon and the momentum problem, we can explain the difference of opinion between Rissanen and Dawid as follows.

Rissanen suggests that we should attempt to avoid poor initial predictions that grossly penalise a complex model. This implies that we should change the construction of the PFSs for the models in stage one (see Figure 4.1). Following his suggestion would have several unappealing consequences however. For one thing, it does not treat models as black boxes (see Section 4.2.2). This rules out using subjective Bayesian priors, even if they are available, and makes models unsuitable as formalisations of predictive strategies by experts. In addition, it does not even seem possible to avoid the momentum phenomenon in all possible model selection scenarios. Consider, for instance, model selection between the Full Bernoulli model and, say, twenty models

like the Single Bernoulli model with parameters that are spread out over the parameter space of the Full Bernoulli model. To avoid the momentum phenomenon between the Full Bernoulli model and every other model in this setting, the Full Bernoulli model should, even for small sample sizes, predict equally well as all the other models. This, however, cannot be achieved.

Dawid, on the other hand, considers the occurrence of the momentum phenomenon perfectly reasonable. He correctly discerns two regions of sample sizes: the initial stages where a simple model is the best predictor of future data, and the region where the data are sufficiently extensive as to demand more detailed description. If we may use our rough sketch from Figure 1 in the preface for illustration, then these regions respectively correspond to regions A and B together and region C. As a consequence of the momentum problem, however, we distinguish between the regions A and B separately. Just like MDL and Bayes factors model selection, Dawid's prequential approach will select the simple model in region B instead of the best-predicting complex model. In addition, by our argument in Section 3.1 region B may get very large. Dawid would therefore be wise to heed Rissanen's warning that the complex model may be grossly penalised for poor initial predictions, which applies exactly to region B.

4.4.2 Predictive Least Squares

The Predictive Least Squares (PLS) principle [Rissanen, 1986b] selects the model that minimises the accumulated squared error of sequentially predicting the next outcome in time-series data. At each time-step it uses the maximum likelihood parameter estimate within each model on the observed data to predict the most likely future outcome. We recognise the momentum phenomenon in a simulation study by Wei [1992], which compares PLS to, among others, AIC [Akaike, 1974] and BIC [Schwarz, 1978] in regression among *normal linear models*. For the current setting there exists a strong relationship [c.f. (1.4) in Wei, 1992] between PLS and predictive MDL, which was introduced in Section 4.1.2. Though the exact relationship between predictive MDL and our approach based on Bayesian universal models has not been completely mapped out in general [De Rooij and Grünwald, 2006], it seems plausible that the momentum phenomenon transfers. The momentum phenomenon might therefore also occur with PLS. Admittedly, this reasoning depends on similarities between procedures that may not (completely) hold true. Remarks in [Wei, 1992], however, strongly remind us of the momentum

phenomenon, which justifies a brief discussion. We will first introduce model selection in regression among normal linear models and then discuss Wei's simulation study.

Regression may be viewed as prediction with side information and is closely related to prediction as considered in this thesis. Normal linear models are of the form:

$$y = \beta' \mathbf{x} + \epsilon \quad \text{with } \epsilon \sim \mathcal{N}(0, \sigma^2),$$

where the independent side information $\mathbf{x}$ is a vector of d variables and the dependent variable y is predicted based on estimates of the d-parameter vector β and the variance σ^2 of the normally distributed noise ϵ. It is assumed that the data arrive in sequence, $(\mathbf{x}^n, y^n) := (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$, and that the noise for each y_i is independent of the other outcomes in the sequence. The task is to predict y_{n+1} given $\mathbf{x}_{n+1}$ and all past observations $(\mathbf{x}^n, y^n)$. The model that estimates all components of β is called the full model. The model set consists of submodels of the full model that exclude different components of $\mathbf{x}$ for the prediction of y, i.e. submodels that always set specific components of β to zero. For normally distributed noise it is natural to measure the loss of the models by the squared error of their predictions. This motivates the PLS principle.

Presumably, the momentum phenomenon and the momentum problem can be reproduced for the regression setting. Suppose, for instance, that the data are sampled from a source in the full model that has most components of β only slightly different from zero. Then we can expect that the submodel that sets these components of β to zero will converge quickly and predict well for small sample sizes, but also that the full model will eventually converge to its optimal parameter estimates and start outperforming the simple submodel. We will then observe the momentum phenomenon and most likely be able to exploit it by the Switch-Point procedure.

Wei's simulation study randomly samples 50 outcomes from a source in different models with $d = 3$ and $\sigma^2 = 1$. The models are of different size: model M_0 estimates one component of β, both models M_1 and M_2 contain M_0 and estimate two components of β, and finally M_3 is the full 3-parameter model, which contains all other models. The experiment is repeated 100 times and the number of times each model is selected is counted. When data are sampled from a source in the relatively small models M_1 and M_2, it is found that PLS selects the generating model 98 and 99 times, respectively, which is highest among all evaluated procedures. When data are sampled from a source in the full model M_3, however, PLS selects the generating model

only 89 times while all of the other procedures select it every time. Wei's examination of the details reveals that in most of the 11 samples for which PLS selected a smaller model, the total loss of model M_3 was dominated by large prediction errors on the first few observations. We conclude that Wei has observed the momentum phenomenon.

Wei considers PLS to be in error on the 11 samples from M_3 on which it did not select the generating model. To avoid overly penalising complex models for poor initial predictions, Wei proposes to exclude the first few observations from the evaluation of predictive accuracy. It remains unclear, however, how many initial observations should be discarded: too many wastes data, too few does not avoid the momentum phenomenon. Wei explains PLS's behaviour by referring to Rissanen [1986b], who also observes PLS's preference for simple models. Rissanen, however, considers such preference desirable, because, for small sample sizes, a small model may better predict future data than a large model that contains the generating source for the data.

The difference in opinion between Wei and Rissanen is illuminated when cast in terms of the momentum phenomenon. If the momentum phenomenon occurs, then it benefits prediction to prefer simple models over more complex models at small sample sizes as claimed by Rissanen. This corresponds to region A in Figure 1. However, the generating more complex model will still become the best predictor eventually, which corresponds to regions B and C from the figure. When this happens it first needs to make up for its poor initial predictions before it is selected. This corresponds to region B. Therefore in region B PLS may still overly penalise complex models for poor initial predictions as assumed by Wei.

4.4.3 Predictive Optimality of the Median Probability Model

In the same regression setting Barbieri and Berger [2004] introduce the *median probability model*, which they define as the model that includes exactly those components of $\mathbf{x}$ as are included by all the models in any set of models of which the total posterior probability is at least 0.5. They call the equivalent of P_{Bayes} in the regression setting the *optimal Bayesian predictor* of y_{n+1} given $\mathbf{x}_{n+1}$. Then they prove that the median probability model, under various assumptions, minimises the expected prediction error under the optimal Bayesian predictor. Subsequently, they claim that this is optimal predictive

model selection. By identifying the momentum problem, however, we have shown that P_{Bayes} is not always the optimal strategy for prediction. It may therefore be questioned whether minimising expected prediction error under P_{Bayes} may be considered optimal. We conclude that, in light of the momentum problem, the claimed predictive optimality of the median probability model should be reevaluated.

4.4.4 Forward Validation

We mention the method of *forward validation* proposed by Hjorth [1982], which may be related to the momentum problem. Forward validation is closely related to Dawid's prequential approach [Rissanen, 1989, p. 68]. Wagenmakers et al. [2006] note that the sequential predictions in forward validation are weighted by the number of observations on which they are based. According to them this "reduces the concern that complex models may be overly penalised in the initial stages of the sequential prediction procedure." Hjorth, however, does not explicitly mention the dependence of the weights on the number of observations. It therefore remains unclear whether forward validation might offer any insights that help to resolve the momentum problem.

4.5 Chapter Summary

We have introduced Dawid's prequential approach to statistics, which asserts that the purpose of statistical inference is the construction of prequential forecasting systems (PFSs). We have interpreted MDL and Bayes as the construction of PFSs in two stages. Then we argued that the design of regular MDL does not take the momentum into account in the second stage, which explains why the Switch-Point procedure, which has been explicitly designed with the momentum phenomenon in mind, is able to improve predictive accuracy compared to regular MDL.

We then explored characteristic properties of the Switch-Point procedure. The Switch-Point procedure is an application of the luckiness principle, which concerns modifications to a code such that it assigns significantly shorter codelength to the data if we are lucky and not much longer codelength if we

are not. We therefore argued that it should always be applied if the occurrence of the momentum phenomenon cannot be ruled out. In the current analysis the gain of the Switch-Point procedure depends on the size of the momentum phenomenon and on the optimal switch-point $\hat{s}$. However, preliminary work suggests that the Switch-Point procedure can be refined such that its gain depends *only* on the size of the momentum phenomenon and not on $\hat{s}$.

Next, we considered implications of our results for model selection. We argued that the Switch-Point model should be considered as an alternative explanation for the data in the Conditional Bernoulli example. Whether it should be considered in the Bernoulli example as well remained unclear.

Finally, we considered prior work related to the momentum problem. A discussion between J. Rissanen and A. P. Dawid was clarified by insights into the momentum problem. Then we recognised the momentum problem in regression with normal linear models, which may be interpreted as prediction with side information. In the same regression setting, we questioned the claimed optimality of the median probability model. Finally, we briefly mentioned forward validation as a candidate to provide further insights into the momentum problem.

4.6 Conclusions

The Minimum Description Length principle equates learning with finding a short description for the data. In this thesis we applied the MDL principle to prediction and model selection when multiple explanations, or models, were available for the data. Prediction is the task of predicting the next outcome given the data; model selection requires to select a single model to explain the data. The resulting MDL procedures could be interpreted as Bayesian Model Averaging and Bayes factors model selection, respectively. Our results therefore transfer to these Bayesian procedures directly.

As our first contribution we identified the *momentum phenomenon* in prediction. The momentum phenomenon arises when one model enables the most accurate predictions of the future given few observations of the past, but predictions based on another model become more accurate when more data are collected. It was proved that the momentum phenomenon may get

very large. We argued, however, that the design of regular MDL does not take the existence of the momentum phenomenon into account.

As our second contribution we therefore developed the Switch-Point procedure, which deals with the momentum phenomenon by adding the Switch-Point model to the set of models in regular MDL. The Switch-Point model represents the hypothesis that the best-predicting model will change over time. We showed that the Switch-Point procedure can never predict much worse than regular MDL, but may predict significantly better when the momentum phenomenon occurs. We argued that for regular MDL the momentum phenomenon should therefore be considered a momentum *problem*. This is the main insight of this thesis.

Finally, we considered adding the Switch-Point model when the task was model selection. We showed that adding the Switch-Point model can never much increase the total codelength for the data, but may reduce the total codelength whenever the momentum phenomenon occurs. By the MDL principle its addition is therefore justified. This is our third contribution.

4.7 Future Work

In this section we will discuss some possible extensions of our results. We first consider connections to research on combining predictions from multiple experts when the relative quality of their predictions is unknown and might change over time; then we suggest some possible approaches to generalising our proofs; and finally we propose an extension of the Switch-Point code that is insensitive to the order of the data.

4.7.1 Tracking the Best Expert

Consider model selection among many nested models, none of which contain the generating source for the data. To guide our thoughts we might, for instance, imagine the models as Markov chains of increasing order and the generating source as some Markov chain of higher order than contained in any of the models. As the larger (higher order) models need more data before they converge to their best predictor and the source for the data is not in any of the models, it is likely that subsequent outcomes are predicted best

by increasingly complex models. In that case the momentum phenomenon would occur at each switch to a more complex model and it should be possible to improve predictive performance by designing an appropriate counterpart to the Switch-Point code.

To this end we might start looking for inspiration in recent work on combining the predictions from multiple experts (models) by [Bousquet and Warmuth, 2002], which we have referred to before in Section 4.2. They consider the (slightly different) problem of sequentially combining predictions from many experts in time-series data when the best-predicting expert shifts back and forth between a few of the experts. Though starting from a different framework, they make efforts to give their approach a Bayesian interpretation in terms of posterior distributions. In addition they connect its efficiency to the codelength of a code that first codes the set of best-predicting experts, then the switch-points, and finally which of the best-predicting experts predicts best in each of the intervals between switch-points.

4.7.2 Extending Proofs

In Section 3.1 we showed that the momentum phenomenon occurs in probability for some generating distributions in the Bernoulli example if a Bayesian universal model with Jeffreys' prior is used as a universal model for the Full Bernoulli model. With few modifications, however, the proof can be extended to cover any continuous prior for the Bayesian universal model. The main difficulty is to show that $f(n_1)$, as defined in (3.14), is of order $o(1)$ or at least $O(1)$. This does not follow from [Balasubramanian, 1997] anymore, because that paper only considers Jeffreys' prior. The required property is explicitly shown in [Grünwald, 2007], however, but no preliminary version of the proof was available at the time of our investigations.

Moreover, our proof in expectation from Section 3.2 suggests that the proof in probability from Section 3.1 might even be extended to general exponential families. The difficulty, then, would lie in finding counterparts to the convex sets E_1 and E_2 that together partition the part of the model where $D(P_\theta \| P_{\theta^*}) \geq h(n_1)$.

4.7.3 Data-ordering Insensitive Switch-Point Code

The Switch-Point code depends on the order of the data, which might be undesirable if there is no natural order to the data. We therefore consider the following extension of the Switch-Point code that is insensitive to the order of the data. Suppose that the data are indexed from 1 to n. Then the Switch-Point code codes the data in exactly that order: $1, \cdots, n$. With equal validity, however, it might have coded the data in any arbitrary fixed order as long as that order did not depend on the data. Let the set of all possible permutations of the n indices be denoted by $\mathcal{O}_n$, let L_o denote the codelength assigned to the data by an alternative Switch-Point code that codes the data in order $o \in \mathcal{O}_n$ and let P_o be the probability distribution such that $-\log P_o(x^n) = L_o(x^n)$. As a first step, consider the Bayesian universal model

$$P_{\mathcal{O}_n}(x^n) = \frac{1}{|\mathcal{O}_n|} \sum_{o \in \mathcal{O}_n} P_o(x^n). \tag{4.3}$$

that uses a uniform prior over the P_o. For data that are independently and identically distributed we would expect the momentum phenomenon to occur for most random reorderings of the data, which implies that most L_o should assign short codelength to the data. Therefore the code with codelengths $L_{\mathcal{O}_n}(x^n) = -\log P_{\mathcal{O}_n}(x^n)$ should assign short codelength to the data as well. Note that this code is insensitive to reordering of the data.

Unfortunately, however, the size of $\mathcal{O}_n$ is prohibitive for all but very small n, which makes computation of $L_{\mathcal{O}_n}(x^n)$ infeasible. We therefore propose the following alternative: select an ordering $o \in \mathcal{O}_n$ uniformly at random. Do this k times for some manageable number k and call the resulting set of orderings $\mathcal{O}_{n,k}$. Then construct the Bayesian universal model

$$P_{\mathcal{O}_{n,k}}(x^n) = \frac{1}{|\mathcal{O}_{n,k}|} \sum_{o \in \mathcal{O}_{n,k}} P_o(x^n).$$

for the set $\mathcal{O}_{n,k}$. As the orderings in $\mathcal{O}_{n,k}$ have been uniformly selected from $\mathcal{O}_n$, most of the orderings in $\mathcal{O}_{n,k}$ should be highly typical and thus exhibit the momentum phenomenon, regardless of the order of the data. The code with codelengths $L_{\mathcal{O}_{n,k}}(x^n) = -\log P_{\mathcal{O}_{n,k}}(x^n)$ should therefore retain the desirable properties of the Switch-Point code while still being insensitive to the ordering of the data. If this code is applied to multiple data sets, then, in order to avoid degenerative performance on the worst data set, it should ideally be reconstructed for every data set by regenerating $\mathcal{O}_{n,k}$.

Bibliography

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974.

V. Balasubramanian. Statistical Inference, Occam's Razor, and Statistical Mechanics on the Space of Probability Distributions. *Neural Computation*, 9:349–368, 1997.

M. M. Barbieri and J. O. Berger. Optimal predictive model selection. *Annals of Statistics*, 32(3):870–897, 2004.

O. Barndorff-Nielsen. *Information and Exponential Families*. John Wiley & Sons, 1978.

A. Barron, J. Rissanen, and B. Yu. The Minimum Description Length Principle in Coding and Modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, October 1998.

A. R. Barron. Information-theoretic Characterization of Bayes Performance and the Choice of Priors in Parametric and Nonparametric Problems. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 6*, pages 27–52. Oxford University Press, 1998.

J. O. Berger and L. R. Pericchi. Objective Bayesian Methods for Model Selection: Introduction and Comparison. In P. Lahiri, editor, *Model Selection*, volume 38 of *Lecture Notes–Monograph Series*, pages 135–193. Institute of Mathematical Statistics, 2001.

O. Bousquet and M. K. Warmuth. Tracking a Small Set of Experts by Mixing Past Posteriors. *Journal of Machine Learning Research*, 3:363–396, 2002.

D. M. Chickering and D. Heckerman. A comparison of scientific and engineering criteria for Bayesian model selection. *Statistics and Computing*, 10:55–62, 2000.

B. S. Clarke and A. R. Barron. Information-Theoretic Asymptotics of Bayes Methods. *IEEE Transactions on Information Theory*, 36(3):453–471, May 1990.

M. A. Clyde. Bayesian Model Averaging and Model Search Strategies. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 6*, pages 157–185. Oxford University Press, 1999.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

I. Csiszár. Sanov Property, Generalized I-Projection and a Conditional Limit Theorem. *The Annals of Probability*, 12(3):768–793, 1984.

A. P. Dawid. Statistical Theory: The Prequential Approach. *Journal of the Royal Statistical Society, Series A*, 147, Part 2:278–292, 1984.

A. P. Dawid. Prequential Data Analysis. In M. Ghosh and P. K. Pathak, editors, *Current Issues in Statistical Inference: Essays in Honor of D. Basu*, Lecture Notes-Monograph Series, pages 113–126. Institute of Mathematical Statistics, 1992a.

A. P. Dawid. Prequential Analysis, Stochastic Complexity and Bayesian Inference. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 4*, pages 109–125. Oxford University Press, 1992b.

S. de Rooij and P. Grünwald. An empirical study of minimum description length model selection with infinite parametric complexity. *Journal of Mathematical Psychology*, 50:180–192, 2006.

I. J. Good. Rational Decisions. *Journal of the Royal Statistical Society. Series B*, 14(1):107–114, 1952.

P. D. Grünwald. *The Minimum Description Length Principle*. The MIT Press, 2007. To appear.

P. D. Grünwald and S. de Rooij. Asymptotic Log-loss of Prequential Maximum Likelihood Codes. In *Learning Theory: 18th Annual Conference on Learning Theory (COLT 2005)*, volume 3559 of *Lecture Notes in Computer Science*, pages 652–667, June 2005.

P. D. Grünwald, I. J. Myung, and M. A. Pitt, editors. *Advances in Minimum Description Length: Theory and Applications*. The MIT Press, 2005.

S. Harnad. The Symbol Grounding Problem. *Physica D: Nonlinear Phenomena*, 42:335–346, June 1990.

U. Hjorth. Model Selection and Forward Validation. *Scandinavion Journal of Statistics*, 9:95–105, 1982.

J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian Model Averaging: A Tutorial. *Statistical Science*, 14(4):382–417, 1999.

R. E. Kass and A. E. Raftery. Bayes Factors. *Journal of the American Statistical Association*, 90(430):773–795, June 1995.

J. L. Kelly, Jr. A New Interpretation of Information Rate. *Bell Systems Technical Journal*, 35:917–926, 1956.

E. E. Leamer. *Specification Searches: Ad Hoc Inference with Nonexperimental Data*. John Wiley & Sons, 1978.

J. Rissanen. Stochastic Complexity and Modeling. *The Annals of Statistics*, 14(3):1080–1100, 1986a.

J. Rissanen. Order Estimation by Accumulated Prediction Errors. *Journal of Applied Probability*, 23:55–61, 1986b.

J. Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15 of *Series in Computer Science*. World Scientific, 1989.

G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978.

E.-J. Wagenmakers, P. Grünwald, and M. Steyvers. Accumulative prediction error and the selection of time series models. *Journal of Mathematical Psychology*, 50:149–166, 2006.

C. Z. Wei. On Predictive Least Squares Principles. *The Annals of Statistics*, 20(1):1–42, March 1992.

APPENDIX A

Posterior Distribution for the Conditional Bernoulli Model

Changing notation for convenience of exposition, (3.67) states that the posterior probability of the next outcome according to the Conditional Full Bernoulli model using Jeffreys' prior is given by

$$P_{\mathcal{M}_3}(Y_{n+1} = 1 | x_{n+1}, x^n, y^n) = \frac{n_1 + \frac{1}{2}}{a + 1},$$

where a denotes the number of occurrences of x_{n+1} in x^n and n_1 denotes the number of $i \in [1, n]$ such that $x_i = x_{n+1}$ and $y_i = 1$. We will now prove this. To this end we first compute the entries of the Fisher information matrix for the Conditional Full Bernoulli model. Then we compute Jeffreys' prior. Letting n_0 denote the number of $i \in [1, n]$ such that $x_i = x_{n+1}$ and $y_i = 0$, which implies that $n_0 + n_1 = a$, we end by showing that

$$P_{\mathcal{M}_3}((X_{n+1}, Y_{n+1}) = (k, 1) | x^n, y^n) = \frac{1}{d} \cdot \frac{n_1 + \frac{1}{2}}{n_0 + n_1 + 1} \tag{A.1}$$

for any k, which implies (3.67).

A.1 Fisher Information

Each source in the Conditional Full Bernoulli model describes the paired observations (x_1, y_1), (x_2, y_2), $\cdots$ as independently and identically distributed.

Therefore

$$\begin{aligned} I_{ij}(\theta^*) &= \lim_{n\to\infty} \frac{1}{n} E_{\theta^*}\left[-\frac{\partial^2}{\partial\theta_i\partial\theta_j} \ln P_\theta(X^n, Y^n)\right]_{\theta=\theta^*} \\ &= E_{\theta^*}\left[-\frac{\partial^2}{\partial\theta_i\partial\theta_j} \ln P_\theta(X, Y)\right]_{\theta=\theta^*} \\ &= E_{\theta^*}\left[-\frac{\partial^2}{\partial\theta_i\partial\theta_j}\left(\ln\frac{1}{d} + Y\cdot\ln\theta_X + (1-Y)\cdot\ln(1-\theta_X)\right)\right]_{\theta=\theta^*}. \end{aligned}$$

Terms in which $X \neq j$ become 0 when taking the derivative to θ_j. Therefore

$$I_{ij}(\theta^*) = \sum_y P_{\theta^*}(j,y)\left[-\frac{d}{d\theta_i}\left(\frac{y}{\theta_j} - \frac{1-y}{1-\theta_j}\right)\right]_{\theta=\theta^*}.$$

It follows that $I_{ij}(\theta^*) = 0$ if $i \neq j$ and if $i = j$, then, as $y \in \{0,1\}$,

$$\begin{aligned} I_{ii}(\theta^*) &= \left[\sum_y P_{\theta^*}(i,y)\frac{y}{\theta_i^2} + \sum_y P_{\theta^*}(i,y)\frac{1-y}{(1-\theta_i)^2}\right]_{\theta=\theta^*} \\ &= \frac{1}{d}\cdot\left[\frac{\theta_i^*}{\theta_i^2} + \frac{1-\theta_i^*}{(1-\theta_i)^2}\right]_{\theta=\theta^*} \\ &= \frac{1}{d}\cdot\frac{1}{\theta_i^*(1-\theta_i^*)}. \end{aligned}$$

A.2 Jeffreys' Prior

Jeffreys' prior can now be computed by

$$\begin{aligned} w_{\text{Jeffreys}}(\theta) &= \frac{\sqrt{|I(\theta)|}}{\int_{\theta\in\Theta}\sqrt{|I(\theta)|}\,d\theta} \\ &= \frac{\sqrt{d^d}\cdot\sqrt{\prod_{i=1}^d 1/\theta_i(1-\theta_i)}}{\sqrt{d^d}\cdot\int_{\theta\in[0,1]^d}\sqrt{\prod_{i=1}^d 1/\theta_i(1-\theta_i)}\,d\theta} \\ &= \frac{\prod_{i=1}^d\sqrt{1/\theta_i(1-\theta_i)}}{\int\cdots\int_0^1\prod_{i=1}^d\sqrt{1/\theta_i(1-\theta_i)}\,\partial\theta_1\cdots\,\partial\theta_d} \\ &= \frac{\prod_{i=1}^d\sqrt{1/\theta_i(1-\theta_i)}}{\prod_{i=1}^d\int_{\theta_i=0}^1\sqrt{1/\theta_i(1-\theta_i)}\,\partial\theta_i} \\ &= \frac{\prod_{i=1}^d\sqrt{1/\theta_i(1-\theta_i)}}{\pi^d}. \end{aligned}$$

A.3 Posterior with Jeffreys' Prior

Let $Z := (X, Y)$ abbreviate our notation. Then

$$P_{\mathcal{M}_3}(x^n, y^n) = P_{\mathcal{M}_3}(z^n) = \int_{\theta\in[0,1]^d} P_\theta(x^n) \cdot P_\theta(y^n|x^n) \cdot w_{\text{Jeffreys}}(\theta)\ d\theta.$$

Let $n_0(i, z^n)$ and $n_1(i, z^n)$ respectively denote the number of occurrences of $(i, 0)$ and $(i, 1)$ in z^n. Then

$$\begin{aligned} P_{\mathcal{M}_3}(z^n) &= \int_{\theta\in[0,1]^d} d^{-n} \cdot \left(\prod_{i=1}^d \theta_i^{n_1(i,z^n)} \cdot (1-\theta_i)^{n_0(i,z^n)}\right) \cdot \frac{\prod_{i=1}^d \sqrt{1/\theta_i(1-\theta_i)}}{\pi^d}\ d\theta \\ &= \frac{d^{-n}}{\pi^d} \cdot \left(\prod_{i=1}^d \int_0^1 \theta_i^{n_1(i,z^n)-\frac{1}{2}} \cdot (1-\theta_i)^{n_0(i,z^n)-\frac{1}{2}}\ d\theta_i\right). \end{aligned}$$

Therefore

$$\begin{aligned} P_{\mathcal{M}_3}((X_{n+1}, Y_{n+1}) = (k,1)|z^n) &= \frac{P(Z_{n+1} = (k,1), z^n)}{P(z^n)} \\ &= \frac{\frac{d^{-(n+1)}}{\pi^d} \cdot \left(\prod_{i\in\{1,d\}-\{k\}} \int_0^1 \theta_i^{n_1(i,z^n)-\frac{1}{2}} \cdot (1-\theta_i)^{n_0(i,z^n)-\frac{1}{2}}\ d\theta_i\right)}{\frac{d^{-n}}{\pi^d} \cdot \left(\prod_{i\in\{1,d\}} \int_0^1 \theta_i^{n_1(i,z^n)-\frac{1}{2}} \cdot (1-\theta_i)^{n_0(i,z^n)-\frac{1}{2}}\ d\theta_i\right)} \\ &\quad \times \int_0^1 \theta_k^{n_1(k,z^n)+1-\frac{1}{2}} \cdot (1-\theta_k)^{n_0(k,z^n)-\frac{1}{2}}\ d\theta_k \\ &= d^{-1} \cdot \frac{\int_0^1 \theta_k^{n_1(k,z^n)+\frac{1}{2}} \cdot (1-\theta_k)^{n_0(k,z^n)-\frac{1}{2}}\ d\theta_k}{\int_0^1 \theta_k^{n_1(k,z^n)-\frac{1}{2}} \cdot (1-\theta_k)^{n_0(k,z^n)-\frac{1}{2}}\ d\theta_k}. \end{aligned}$$

Now define

$$F_\theta(n_0, n_1) := \int_0^1 \theta^{n_1-\frac{1}{2}} \cdot (1-\theta)^{n_0-\frac{1}{2}}\ d\theta.$$

Then this can be rewritten as

$$P_{\mathcal{M}_3}((X_{n+1}, Y_{n+1}) = (k,1)|z^n) = \frac{d^{-1} F_{\theta_k}(n_0(k,z^n), n_1(k,z^n)+1)}{F_{\theta_k}(n_0(k,z^n), n_1(k,z^n))}.$$

It will be convenient to show that

$$\begin{aligned} F_\theta(n_0, n_1) &= \int_0^1 (1-\theta+\theta) \cdot \theta^{n_1-\frac{1}{2}} \cdot (1-\theta)^{n_0-\frac{1}{2}}\ d\theta \\ &= \int_0^1 \theta^{n_1-\frac{1}{2}} \cdot (1-\theta)^{n_0+\frac{1}{2}}\ d\theta + \int_0^1 \theta^{n_1+\frac{1}{2}} \cdot (1-\theta)^{n_0-\frac{1}{2}}\ d\theta \\ &= F_\theta(n_0+1, n_1) + F_\theta(n_0, n_1+1). \end{aligned}$$

In addition, integration by parts shows that

$$F_\theta(n_0+1,n_1)=\frac{n_0+\frac{1}{2}}{n_1+\frac{1}{2}}F_\theta(n_0,n_1+1).$$

We now abbreviate $n_0(k,z^n)$ to n_0 and $n_1(k,z^n)$ to n_1. Then

$$\begin{aligned}P_{\mathcal{M}_3}((X_{n+1},Y_{n+1})=(k,1)|z^n)&=\frac{d^{-1}F_{\theta_k}(n_0,n_1+1)}{F_{\theta_k}(n_0,n_1)}\\&=\frac{d^{-1}F_{\theta_k}(n_0,n_1+1)}{F_{\theta_k}(n_0+1,n_1)+F_{\theta_k}(n_0,n_1+1)}\\&=\frac{d^{-1}F_{\theta_k}(n_0,n_1+1)}{\frac{n_0+\frac{1}{2}}{n_1+\frac{1}{2}}F_{\theta_k}(n_0,n_1+1)+F_{\theta_k}(n_0,n_1+1)}\\&=d^{-1}\cdot\frac{n_1+\frac{1}{2}}{n_0+n_1+1},\end{aligned}$$

which completes the proof of (A.1).