Department of Computer Science
Series of Publications A
Report A-2013-8

Normalized Maximum Likelihood Methods for Clustering and Density Estimation

Panu Luosto

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium XIV, on 14th September 2013 at 10 a.m.

University of Helsinki
Finland

Supervisor
Jyrki Kivinen, University of Helsinki, Finland

Pre-examiners
Wray Buntine, NICTA and The Australian National University
Ioan Tabus, Tampere University of Technology, Finland

Opponent
Jaakko Peltonen, Aalto University, Finland

Custos
Jyrki Kivinen, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: postmaster@cs.helsinki.fi
URL: http://www.cs.Helsinki.fi/
Telephone: +358 9 1911, telefax: +358 9 191 51120

ISSN 1238-8645
ISBN 978-952-10-9179-7 (paperback)
ISBN 978-952-10-9180-3 (PDF)
Computing Reviews (1998) Classification: G.3, H.1.1, I.2.6
Helsinki 2013
Helsinki University Print

Normalized Maximum Likelihood Methods for Clustering and Density Estimation

Panu Luosto

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland

PhD Thesis, Series of Publications A, Report A-2013-8
Helsinki, September 2013, 67 + 67 pages
ISSN 1238-8645
ISBN 978-952-10-9179-7 (paperback)
ISBN 978-952-10-9180-3 (PDF)

Abstract

The normalized maximum likelihood (NML) distribution has an important position in minimum description length based modelling. Given a set of possible models, the corresponding NML distribution enables optimal encoding according to the worst-case criterion. However, many model classes of practical interest do not have an NML distribution. This thesis introduces solutions for a selection of such cases, including for example one-dimensional normal, uniform and exponential model classes with unrestricted parameters. The new code length functions are based on minimal assumptions about the data, because an approach that would be completely free of any assumptions is not possible in these cases. We also use the new techniques in clustering, as well as in density and entropy estimation applications.

Computing Reviews (1998) Categories and Subject Descriptors:

G.3	Probability and Statistics
H.1.1	Systems and Information Theory
I.2.6	Learning: parameter learning

General Terms:
statistical modelling, data analysis, machine learning

Additional Key Words and Phrases:
information theory, minimum description length, clustering, density estimation

Acknowledgements

I want to thank my supervisor, Jyrki Kivinen, for guiding me through the process of writing this thesis. I am also grateful to my mentor Heikki Mannila, who gave some of the very first research ideas. I thank the pre-examiners Wray Buntine and Ioan Tabus for their constructive comments that helped to improve the thesis. Naturally, warm thanks belong also to my co-writers Petri Kontkanen, Ciprian Giurcăneanu, and Kerkko Luosto.

Original Publications and Contributions

This doctoral dissertation is based on the following five publications, which are referred to in the text as Papers I–V. The papers are reprinted at the end of the thesis.

Paper I: Panu Luosto. Code lengths for model classes with continuous uniform distributions. In *Proceedings of the Workshop on Information Theoretic Methods in Science and Engineering*, 2010.

Paper II: Panu Luosto, Jyrki Kivinen, and Heikki Mannila. Gaussian clusters and noise: an approach based on the minimum description length principle. In *Proceedings of the 13th International Conference on Discovery Science*, pages 251–265, 2010.

Paper III: Panu Luosto and Petri Kontkanen. Clustgrams: an extension to histogram densities based on the minimum description length principle. *Central European Journal of Computer Science*, 1(4):466–481, 2011.

Paper IV: Panu Luosto, Petri Kontkanen, and Kerkko Luosto. The normalized maximum likelihood distribution of the multinomial model class with positive maximum likelihood parameters. University of Helsinki, Department of Computer Science, Technical Report C-2012-5, 2012.

Paper V: Panu Luosto, Ciprian Doru Giurcăneanu, and Petri Kontkanen. Construction of irregular histograms by penalized maximum likelihood: a comparative study. In *Proceedings of the IEEE Information Theory Workshop*, pages 297–301, 2012.

In Papers I, III and IV we develop new NML based criteria for model selection. From these three papers only Paper III contains a limited number of experimental tests. Papers II and V are somewhat more application-oriented. The main contributions of all the papers are listed below.

Paper I: We introduce normalized maximum likelihood based code lengths for model classes with uniform distributions, considering arbitrary one and two-dimensional balls, and origin-centred balls in any dimension. Our main contribution is to generalize the NML codes for situations where the range of the maximum likelihood parameters is not known before seeing the data.

Paper II: We derive an NML based code length for mixtures with several normal components and one uniform component. We also introduce a search heuristic for finding the best clustering of this type with an unknown number of clusters.

Paper III: We extend the idea of a histogram by allowing a wider selection of components other than just the uniform one. The approach is based on clustering, and the component type is selected using an NML based criterion.

Paper IV: We calculate the NML distribution of a multinomial model class in the restricted case when it is known that the maximum likelihood parameters are always positive. This case is relevant for clustering applications.

Paper V: We compare empirically different penalized maximum likelihood based methods for density and entropy estimation. One of the NML based methods is a novel variant of the MDL histogram introduced in [18]. We also examine how the risk bounds used by statisticians can be applied to the original MDL histogram.

The contribution of the present author is substantial in all the five papers, and he is the main contributor in Papers I–IV.

In Paper IV, the proof of Theorem 2 is by Kerkko Luosto. In Paper V, the idea of the new variant of the MDL histogram (NML-2) is due to the current author, who is also responsible for the experimental evaluation of the methods, described in Section 5 of the paper.

After the publication of Paper IV, it came to the knowledge of the authors that Theorem 1 was proved earlier in [26] (see Addendum to Paper IV).

Contents

Chapter 1

Introduction

Imagine that you should describe a mushroom growing in the nearby forest to someone else in writing. If you and the other person know the local fungi well enough, it would be practical and effective to start the description by stating the species of the fungus. The word pair "Russula paludosa"[1] does not tell what exact shape, size or colour the mushroom has, but once you know the species, you have an idea how the mushroom is likely to look. Also, describing individual details of a specific mushroom is then much easier than starting from scratch. A correct determination of the species is not necessarily needed, because a mere resemblance with the stated species makes the description simpler and shorter.

This thesis is about finding the quintessence from data, using information theory [7] and more specifically the *minimum description length* (MDL) principle [11, 35, 36] as a tool. The phrasing should be understood in a very specific context, in which the meaning of the word "quintessence" has a well-defined and quite commonplace character. In this thesis, numerical data sets are fungi, and probability distributions are names of fungus species. We associate data with a distribution that enables a short description of the data points. The matching distribution, called the *model*, is the quintessence of the data sample. The model alone does not specify any particular point of the data, but it implies how the data set is likely to be. The logics of the MDL principle do not require an assumption that the data were indeed generated by some distribution. The model is just the essential information that can be extracted from the data within our framework. With that knowledge we can then accomplish such practical tasks as clustering and density estimation, which are the subject of Chapter 4.

We can sharpen the illustration with mushrooms and probability distri-

[1] In Finnish punahapero.

butions a little. Think now of a set of probability distributions corresponding to a name for a mushroom species. Such a set is called a *model class* and its elements are typically distributions of a same type but with different parameters. In the mushroom world, equivalents for the parameters could be weight and dimensions of a certain mushroom species. All the finer details – irregularities and worm holes – must be described in another form, as they cannot be captured by a few numbers only. To use statistical language, they are random. The so called *model selection problem*[2], is about choosing a suitable model class. That is useful knowledge about the data, as it is useful to recognize the species of a mushroom in a forest.

In our mushroom example, it is essential to choose a correct or an otherwise matching species name to keep the textual description of the mushroom short. The number of known fungus species is naturally finite, but there are infinitely many sets with probability distributions. It is important to notice that the model selection process is feasible only when the sets of possible model classes and models are restricted in a suitable way. If little is known about the data, the first question is what kind of distributions we should consider? The answer is very case-dependant, and as such out of the scope of the thesis. However, the MDL principle can help us to choose a suitable model class from a heterogeneous selection. For example, our *clustgram* (Section 4.2) generalizes the common histogram and provides a method for selecting the best matching distribution types for a mixture density from a small fixed selection.

A common issue in model selection arises in a situation where the model classes have different complexities, like mixtures with two or twenty components. A naive maximum likelihood method overfits, that is, it favours complex models and is thus useless for applications. The complexity of a model class can be penalized easily, but it should not be done arbitrarily since we want to avoid underfitting as well. The MDL has proven to be a valuable criterion for model selection. Especially, it seems to handle the uncertainty caused by small samples in a satisfactory way.

Normalized maximum likelihood (NML) is one of the most important MDL related techniques. It has important optimality properties, but straightforward computations of NML code lengths are sometimes unfeasible in practice. Section 3.1 shows how a complex sum needed for a particular NML distribution can be calculated efficiently after suitable mathematical manipulations. The NML can often be succesfully approximated as well, but we do not discuss approximation methods in this thesis.

A central problem for the thesis are model classes that do not have

[2] Or more correctly: model class selection problem.

NML distributions. Computing an NML code length involves calculating a normalizing term, which measures the complexity of a model class. If the model class is rich enough, the normalizing term can be infinite, and there is no corresponding NML distribution. One option in this situation is to redefine either the model class or the set of possible data sequences. Even if a true NML probability measure could be achieved this way, the approach may be problematic in practice. In particular, if we have only little prior information about the data, arbitrary assumptions may have a strong effect on the code lengths. Section 2.5 illustrates the problem by examples.

An alternative solution to the problem of infinitely complex model classes is to keep the model class and domain of the data unaltered and to choose another coding method having roughly similar good properties as the NML distributions. Our contributions to the topic are presented in Section 3.2. We develop a method that performs well with the data we consider probable. In turn we accept slowly weakening performance when the data gets more unlikely. Prior information about the data is thus needed but the data analyst is given more flexibility when crucial assumptions about the data and models are made.

Clustering is a natural application for our NML based methods, which make it possible to find correspondences between model classes and clustering structures. The basic idea is to find a good clustering for every model class, and then to pick the clustering having the shortest code length. We say "a good clustering" instead of "optimal" here, because it is typically very hard to find an optimal solution given a model class. The one-dimensional case forms an exception under certain conditions, that is why we pay special attention to it in Chapter 4. We extend traditional histograms to clustgrams with several component types, and use them for one-dimensional density and entropy estimation.

The thesis is structured as follows. In Chapter 2, which serves as a technical introduction, model selection using normalized maximum likelihood is described and a central research topic, the infinite parametric complexity, is presented. The next two chapters relates to the main contribution of the thesis: in Chapter 3 we discuss code word lengths for certain model classes, and in Chapter 4 we introduce clustering and density estimation applications that utilize the new code word lengths. Chapter 5 includes concluding remarks.

Chapter 2

Model Selection with Normalized Maximum Likelihood

This chapter introduces concepts and techniques that are central for the thesis. We start by describing the model selection problem as it is encountered in our work. After a short introduction to the minimum description length principle in Section 2.2, we discuss in Section 2.3 the normalized maximum likelihood, which is one of the most important constructs in modern MDL research. Section 2.4 returns to the model selection problem, now from the NML point of view. In Section 2.5, we portray a difficult problem relating to the application of the NML in many cases: the infinite parametric complexity. We conclude the chapter by outlining various approaches to this major research problem in Section 2.6. Our contributions related to the infinite parametric complexity are discussed in the next chapter.

2.1 Model Selection Problem

Let $D \subset (\mathbb{R}^d)^n$ be the domain of the data. Here d is the dimension of a data point, and n is the number of data points. We call the set $M = \{p(\cdot;\, \theta) \mid \theta \in \Theta\}$ a *model class*, where $\Theta \subset \mathbb{R}^k$ is a parameter space and $p(\cdot;\, \theta)$ is a probability measure with parameter vector θ and support D. Additionally, we call the possibly finite collection of model classes $\mathcal{M} = \{M_1, M_2, \dots\}$ a *model family*. Given a data sample $\mathbf{x} \in D$, a fundamental problem is to select from $\mathcal{M}$ the model class that is the most plausible description of the properties of $\mathbf{x}$. Once the model class is chosen, the preferred model (probability measure) for $\mathbf{x}$ is usually the one that maximizes the probability. The combined problem of model class and model selection is commonly called simply the *model selection problem.*[1]

The previous terminology is somewhat loosely defined to be useful as such. Therefore we present next an example with normal mixtures, which we shall use in this and later sections for illustrating some typical difficulties in model selection. We point out that Paper II concentrates on hard clustering of normal mixtures, but in the presence of an additional uniform noise component. As a part of a more versatile context, also Paper III comes close to the problematics of clustering normal mixtures.

We start by defining the product densities of one-dimensional normal mixtures with 1 to N components. In this case, the elements of a data vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ are scalars. Let $\mu_a, \mu_b \in \mathbb{R}$, $\sigma_a^2, \sigma_b^2 \in \left]0, \infty\right[$, and for all $k \in \{1, 2, \ldots N\}$, let the parameter space be $\Theta_k = \Delta_{k-1} \times [\mu_a, \mu_b]^k \times [\sigma_a^2, \sigma_b^2]^k$, where $\Delta_{k-1} = \{(w_1, \ldots, w_k) \in \left]0, 1\right]^k \mid \sum_{j=1}^k w_j = 1\}$. (Restricting the mean and the variance may seem somewhat arbitrary here. It is indeed a problematic subject, and we explain its relevance for the NML in Section 2.5.) The definition of a mixture with k components is now $p_k : \mathbb{R}^n \to \left]0, \infty\right[$,

$$p_k(\mathbf{x}; \theta_k) = \prod_{i=1}^{n} \sum_{j=1}^{k} w_j \, \varphi(x_i;\, \mu_j, \sigma_j^2)\,,$$

where $\theta_k = (w_1, \ldots, w_k, \mu_1, \ldots, \mu_k, \sigma_1^2, \ldots \sigma_k^2) \in \Theta_k$ is a parameter vector and $\varphi(\cdot;\, \mu, \sigma^2)$ is the density function of a one-dimensional normal distribution with mean μ and variance σ^2.

Now we can define for all $k \in \{1, 2, \ldots, N\}$ a model class $M_k = \{p_k(\cdot;\, \theta) \mid \theta \in \Theta_k\}$. In this context, a classical machine learning problem is to ask: What is the most plausible model class $M_k \in \{M_1, M_2, \ldots, M_N\}$ for the data $\mathbf{x} \in \mathbb{R}^n$, and which distribution in M_k fits the data best? It is well-known that simply picking the pair (k, θ) that maximizes $p_k(\mathbf{x};\, \theta)$ leads to over-fitting, i.e., the chosen model class has too many components. "Too many" is not only an intuitive concept. The misbehaviour of the naive strategy can be verified for example by producing a data sample according to a known mixture of normals, and comparing then the structures of the original source and the recovered mixture. In this case when the source is known, we can also examine how far the recovered mixture density is from the source using such measures as Kullback-Leibler distance or squared Hellinger distance.

If M_k is given, it is reasonable to maximize the likelihood, that is, to choose the parameter vector θ that maximizes density $p_k(\mathbf{x};\, \theta)$. Therefore,

[1] The reader should be aware that some writers refer to M using the term "model" and call $\mathcal{M}$ a "model class".

most methods for model class selection rely on a function $C(M_k)$ with which the maximum likelihoods are scaled. The strategy is then to select the $k \in \{1, 2, \ldots, N\}$ maximizing

$$C(M_k) \cdot p_k(\mathbf{x};\, \hat{\theta}_{M_k}(\mathbf{x})) \tag{2.1}$$

where $\hat{\theta}_{M_k}(\mathbf{x})$ is the maximum likelihood parameter vector and thus

$$p_k(\mathbf{x};\, \hat{\theta}_{M_k}(\mathbf{x})) = \max\{p_k(\mathbf{x};\, \theta) \mid \theta \in \Theta_k\}\,.$$

Model class selection criteria of the form (2.1) can be called *penalized maximum likelihood* (PML) methods. The Akaike information criterion [2] and the Bayesian information criterion (BIC) [41] are two well-known examples, both from the 1970s. The normalized maximum likelihood is a modern PML method for model class selection with appealing optimality properties, and it is in the focus of this thesis. Paper V compares empirically the behaviour of five different PML methods in one-dimensional density and entropy estimation. Three of the methods are based on the NML, one is a variant of the BIC, and one has its roots in an approach that strives to minimize the upper bound for the statistical risk.

2.2 Minimum Description Length Principle

The minimum description length (MDL) principle [11, 35, 36] has evolved over the years [29, 32, 31], but in its all forms, it is based on the same idea: finding useful information in the data is equated with compressing the data, as only regularities of the data make compression possible. The expressions "useful information" and "regularities of the data" have to be considered with regard to a collection of models, that is, probability measures. How to choose an appropriate selection of models is up to a human expert, because the MDL theory only operates within a given framework of model classes. On the other hand, it is exactly this limitation that makes the MDL principle a practical machine learning tool. Solomonoff's *algorithmic information theory* [44, 45, 23] treats learning in its most general form, where the objective is to find the shortest computer program that outputs the given data sample. However, it is easy to prove that the problem is uncomputable. Even in its restricted and feasible forms, the generality of the algorithmic information theory makes its practical use very difficult.

Partly because of historical reasons, the term “MDL principle” is used in connection with quite different techniques. However, the key idea of the MDL principle is to find the optimal way of communication with the help of a model class collection (we explain the optimality criterion in the next section). First, we determine an optimal code for each model class. Then, given a sample of data, we choose the model class that is associated with the shortest code length for the sample. When the main interest is model selection, not real communication, it is sufficient to know the lengths of the code words, and there is no need to choose the actual code words.

Coding according to a fixed distribution is always based on a fundamental result of information theory. Let p be a probability mass function. According to Kraft’s inequality [7, Section 5.2], a valid code exists when the code word length for every possible data sequence $\mathbf{x}$ is $-\log p(\mathbf{x})$, assuming for simplicity that this choice yields integer code lengths. Those lengths are also optimal. More precisely, they minimize the expected code word length when the expectation is taken over the distribution p. The average word length equals then the entropy of the distribution. The negative logarithms of probabilities are not necessarily integers, but it is easy to choose the lengths so that there is at most one bit overhead in expectation compared to the entropy [7, Section 5.4], which is always the lower bound. For our modelling purposes, the integer constraint is not relevant, and we always simply call $-\log p(\mathbf{x})$ a code length.

We also follow a common convention of calling negative logarithms of the densities code lengths, which is reasonable if the densities of a model class are well-behaving enough. In practice, problematic densities are seldom used; we give in the following some intuition about the situation. For concrete communication according to a density, continuous values of the data domain have to be discretized first. For example, set $(\mathbb{R}^d)^n$ can be partitioned into half-open hypercubes each having a volume ϵ, the centre of a hypercube being the approximation for any point inside it. Let $\mathbf{x}$ be a centre of such a hypercube. The ideal code word length for $\mathbf{x}$ would be the negative logarithm of the probability mass P in the hypercube. But if volume ϵ is small enough and the density f in question is continuous, the quantity $-\log(f(\mathbf{x}) \cdot \epsilon) = -\log f(\mathbf{x}) - \log \epsilon$ is in turn a good approximation of $-\log P$. When comparing which density of two alternatives produces the shorter code for $\mathbf{x}$, the constant term $-\log \epsilon$ can be thus ignored. Also such discontinuity that occurs at the edges of the domain of a uniform distribution over an interval, is not a problem for the terminology, because we can still discretize the interval in a sensible way.

2.3 Normalized Maximum Likelihood

The normalized maximum likelihood (NML) code, sometimes also called the Shtarkov code [43], has a special position in MDL research because of its optimality properties. We shall give the basic optimality property below, for a more thorough discussion, see [34]. We start by giving the definition of the NML distribution when the model class consists of probability mass functions, and the data are discrete. Let $M = \{p(\cdot;\, \theta) \mid \theta \in \Theta)\}$ be a model class, where $p(\cdot;\, \theta) : D \to]0, 1]$ is a probability mass function for all $\theta \in \Theta$. We consider here only encodable data, or data points in set D. Let $\hat{\theta}_M : D \to \Theta$ be the maximum likelihood parameter estimator.

The shortest code length for $\mathbf{x} \in D$ according to any member of M is $-\log p(\mathbf{x};\, \hat{\theta}_M(\mathbf{x}))$, because $\hat{\theta}_M(\mathbf{x})$ maximizes the likelihood by definition. The mapping $\mathbf{x} \mapsto p(\mathbf{x};\, \hat{\theta}_M(\mathbf{x}))$ cannot be used for encoding for the simple reason that it is not a density function unless the case is trivial. But it still makes sense to compare the *regret*, i.e., the difference $\mathrm{REG}_M(q, \mathbf{x}) = -\log q(\mathbf{x}) - (-\log p(\mathbf{x};\, \hat{\theta}_M(\mathbf{x}))$ between the code length of $\mathbf{x}$ according to probability mass function q and the shortest possible code length according to any element in M. All forms of the MDL principle strive to minimize the regret. Because the MDL principle deliberately avoids any assumptions about how the data were actually generated, this means specifically minimizing the worst-case regret. The minimization problem is well-defined if the sum

$$C(M, D) = \sum_{\mathbf{x} \in D} p(\mathbf{x};\, \hat{\theta}_M(\mathbf{x})) \tag{2.2}$$

is finite. It is easy to see that the solution is then the normalized maximum likelihood

$$p^{\mathrm{NML}}_{M,D}(\mathbf{x}) = \frac{p(\mathbf{x};\, \hat{\theta}_M(\mathbf{x}))}{C(M, D)}\,,$$

because for all $\mathbf{x} \in D$ regret $\mathrm{REG}_M(p^{\mathrm{NML}}_{M,D}, \mathbf{x}) = \log C(M, D)$ is constant. We can formulate that

$$\inf_{q \in Q} \sup_{\mathbf{x} \in D} \mathrm{REG}_M(q, \mathbf{x}) = \log C(M, D)\,, \tag{2.3}$$

where Q is the set of all density functions that are defined in D. The NML probability is called *minimax optimal*, as it is the minimizing distribution in (2.3). Given M and D, the code length $-\log p^{\mathrm{NML}}_{M,D}(\mathbf{x})$ is called the *stochastic complexity* of $\mathbf{x}$, and the quantity $\log C(M, D)$ is called the *parametric complexity*. Hence, the stochastic complexity describes the complexity of

$\mathbf{x}$ in this context, whereas the parametric complexity is a property of the model class only.

The NML can be generalized for densities in a straightforward way by replacing the sum in (2.2) with an integral. In that case, let $M = \{f(\cdot;\, \theta) \mid \theta \in \Theta\}$ be a model class, where $f(\cdot;\, \theta) : D \to\,]0, \infty[$ is a density function for all $\theta \in \Theta$. Then the normalization term for the NML density is

$$C(M, D) = \int_D f(\mathbf{x};\, \hat{\theta}(\mathbf{x}))\,. \tag{2.4}$$

Unfortunately, all model classes do not have corresponding NML distributions, and even if the distribution exists, its calculation or approximation may be difficult. Our example with mixtures of normal distributions from Section 2.1 illustrates another notable problem of the NML. Also in this case the calculation of the NML is very difficult, because finding the maximum likelihood parameters for a particular data vector in a multi-component mixture is already problematic. But in addition to that, defining the parameter space Θ_k using μ_a, μ_b, σ_a^2, and σ_b^2 as in our example is not possible in many real-world applications without making arbitrary decisions. These problems and some solutions to them are discussed in Sections 2.5 and 2.6 as well in Papers I, II and III.

2.4 Model Class Selection by Complete MDL

We saw in the previous section that given a model class with a finite parametric complexity, the corresponding NML distribution is the most effective encoding method in the worst-case sense. It is also easy to see how the NML can be used for model class selection according to the penalized maximum likelihood criterion in (2.1). However, the criterion in (2.1) is in its general form an ad hoc method, and until this point, we have not given a precise rationale behind the use of NML for model class selection.

The term *complete minimum description length*[2] [36] refers to the code length in NML encoding, either in the connection to a "normal" model class as in the previous section, or corresponding to a model class with NML distributions. The latter case shows how the information theoretic goal of minimizing the code length in the worst case can be applied to the model class selection problem.

We start with some familiar definitions. Let the domain of the data be D. Let $\mathcal{M} = \{M_k \mid k \in I\}$ be a family of model classes, where $I \subset \mathbb{N}$ is an

[2] In contrast, *incomplete* or *general MDL* refers in [36] to the older, more general MDL principle as defined in [29].

index set and for all $k \in I$ model class $M_k = \{f_k(\cdot;\, \theta) \mid \theta \in \Theta_k\}$ consists of probability density functions. We let $\hat{\theta}_k : D \to \Theta_k$ denote the ML parameter estimator for all $k \in I$. In order to simplify our notation, we write the NML density according to the model class M_k as

$$\hat{f}(\mathbf{x};\, k) = \frac{f_k(\mathbf{x};\, \hat{\theta}_k(\mathbf{x}))}{C(M_k, D)} \tag{2.5}$$

where $C(M_k, D) = \int_D f_k(\mathbf{y};\, \hat{\theta}_k(\mathbf{y}))$. We assume that the NML distributions exist, that is, parametric complexity $C(M_k, D)$ is finite for all $k \in I$.

Let us consider the NML densities from a communication point of view. The definition in (2.5) provides us with a code book for each model class M_k, and given data $\mathbf{x}$, it seems reasonable to select the model class, the code book of which contains the shortest code word length for $\mathbf{x}$. However, if the receiver does not know which code book to use, the communication scheme is still incomplete. There is a correspondence with the problem of choosing a code given a model class M_k. We had an optimal parameter estimator $\hat{\theta}_k$ for all $k \in I$, and the NML density gave us the worst-case optimal code. Now, given the model family $\mathcal{M}$ and data $\mathbf{x}$, we determine a code that uses all the code books in an worst-case optimal way.

We start by defining a model class of $M' = \{\hat{f}(\cdot;\, k) \mid k \in I\}$ containing the previously defined NML distributions. We define the corresponding ML parameter estimator as

$$\hat{k}(\mathbf{x}) = \min\left\{k \mid k \in I,\ \hat{f}(\mathbf{x};\, k) = \max\{\hat{f}(\mathbf{x};\, j) : j \in I\}\right\}. \tag{2.6}$$

Assume that $\hat{k}$ is well-defined and that

$$\begin{aligned} C(M', D) &= \int_D \hat{f}(\mathbf{y};\, \hat{k}(\mathbf{y})) && (2.7)\\ &= \sum_{j \in I} \int_{\{\mathbf{y} \,\mid\, \mathbf{y} \in D,\ \hat{k}(\mathbf{y}) = j\}} \hat{f}(\mathbf{y}; j) && (2.8) \end{aligned}$$

is finite. Then it is possible to define an NML distribution corresponding to the model class M'. Its density function is

$$\hat{f}(\mathbf{x}) = \frac{\hat{f}(\mathbf{x};\, \hat{k}(\mathbf{x}))}{C(M', D)}. \tag{2.9}$$

In this case, an optimal model class selector exists, and it is indeed $\hat{k}$.

Notice that if I is finite, we see from (2.8) that $C(M', D) \le |I|$, and therefore $\int_D (1/|I| \cdot \hat{f}(\mathbf{y};\, \hat{k}(\mathbf{y}))\,) \le 1$. Thus a two-part code, in which code book index k is encoded with a uniform code and data $\mathbf{x}$ with $\hat{f}(\cdot;\, k)$, is usually inefficient. However, for our model selection problem the actual value of parametric complexity $C(M', D)$ has no effect as long it is finite.

If $C(M', D)$ is infinite, a constant regret cannot be achieved, and the situation is much more complicated. We can assume $I = \mathbb{N}$. The most straightforward solution is still to use $\hat{k}$ for model selection. A simple corresponding coding strategy would be to choose the code lengths according to the density

$$g(\mathbf{x}) = p(\hat{k}(\mathbf{x})) \frac{\hat{f}(\mathbf{x};\, \hat{k}(\mathbf{x}))}{\int_{\{\mathbf{y} | \mathbf{y} \in D,\, \hat{k}(\mathbf{y}) = \hat{k}(\mathbf{x})\}} \hat{f}(\mathbf{y};\, \hat{k}(\mathbf{x}))}\,, \tag{2.10}$$

where $p : \mathbb{N} \to [0, 1]$ is a suitably chosen probability mass function.

In (2.10) we have as a coefficient an NML density on the condition that $\hat{k}(\mathbf{x})$ is known. Unfortunately, calculating the normalizing integral can be unfeasible in practice, making it very difficult to quantify the regret. But it is easy to see that we have an upper bound $\mathrm{REG}_M(g, \mathbf{x}) \leq -\log p(\hat{k}(\mathbf{x}))$.

If we can control the upper bound so that the regret is fairly uniform and not too large in the set of essential model classes (defined e.g. by prior knowledge), choosing the model class simply by $\hat{k}(\mathbf{x})$ might be justified. A reasonable candidate for p is Rissanen's prior for integers [30]. It decreases however quite unevenly with first few values and assigns a relatively large amount of probability mass for the small k. Luckily, fixing these shortcomings for practical applications is not difficult. The main topics of this thesis do not include infinite model families, but Section A.1 contains the derivation of a probability mass function that is closely related to Rissanen's prior for integers but without its minor faults.

Let us consider lastly a simple model selection strategy that is based on maximizing the density $g_1(\mathbf{x};\, k) = p(k)\hat{f}(\mathbf{x};\, k)$. This two-part code is inefficient in the normal case where one data sequence has non-zero densities in several model classes.

Roos et. al report in [38] that using just $\hat{k}$ with a very large number of model classes led to poor results an image denoising application. However, the most important thing in that particular case was probably that the model family had a natural inner structure. Taking it into account in the coding improved the performance of the denoising algorithm. The structure of the uppermost model family was $\mathfrak{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n\}$, and for all model class indices i such that $M_i \in \mathcal{M}_j \in \mathfrak{M}$ the authors let $p(i) = 1/n \cdot 1/|\mathcal{M}_j|$, thus penalizing model classes in large model families. This can be seen as a simplified alternative of determining the NML distributions for each $\mathcal{M}_j \in \mathfrak{M}$.

2.5 Infinite Parametric Complexity

A notable problem by the use of NML for model selection is that many model classes of practical interest have an infinite parametric complexity, and they thus lack a corresponding NML distribution. In the previous section, we considered a special example of such a problem: the model class consisted of NML distributions and the only parameter to be estimated was the index of a distribution. If the distributions in a model class have continuously valued parameters, we have in practice more options to handle the problem. For example, it is sometimes possible to find a similar but less general model class that suits our purposes, or to restrict the domain of the data. In this section, we illustrate these simple options using an example with geometric distributions. For an example with one-dimensional normal distributions, see [10]. Section 2.6 is devoted to more advanced methods.

In our example we discuss product densities of n independent and identically-distributed geometric random variables. Let $\mathbb{N}_+ = \{1, 2, \dots\}$ and let the probability of a sequence $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{N}_+^n$ with the parameter $\theta \in \,]0, 1]$ be $p(\mathbf{x};\, \theta) = \prod_{i=1}^n (1-\theta)^{x_i - 1}\theta$ (we use here the notational convention $0^0 := 1$). Let the model class be $M = \{p(\cdot;\, \theta) \mid \theta \in \,]0, 1]\}$. Given $\mathbf{x}$, the ML estimate of the parameter θ is $\hat{\theta}_M(\mathbf{x}) = n / \sum_{i=1}^n x_i$. We get a lower bound of the normalization sum for M as follows:

$$\begin{aligned}
C(M, \mathbb{N}_+^n) &= \sum_{\mathbf{x} \in \mathbb{N}_+^n} p(\mathbf{x}; \hat{\theta}_M(\mathbf{x})) && (2.11)\\
&= \sum_{\mathbf{x} \in \mathbb{N}_+^n} \prod_{i=1}^n \left(1 - \frac{n}{\sum_{i=1}^n x_i}\right)^{x_i - 1} \frac{n}{\sum_{i=1}^n x_i} \\
&= \sum_{s=n}^{\infty} \sum_{\substack{\mathbf{x} \in \mathbb{N}_+^n, \\ \sum_{i=1}^n x_i = s}} \left(1 - \frac{n}{s}\right)^{s-n} \left(\frac{n}{s}\right)^n \\
&= \sum_{s=n}^{\infty} \binom{s-1}{n-1} \left(1 - \frac{n}{s}\right)^{s-n} \left(\frac{n}{s}\right)^n \\
&\geq \sum_{s=n}^{\infty} \left(\frac{s-1}{n-1}\right)^{n-1} \left(1 - \frac{n}{s}\right)^{s-n} \left(\frac{n}{s}\right)^n \\
&= \frac{n^n}{(n-1)^{n-1}} \sum_{s=n}^{\infty} \left(\frac{s-1}{s}\right)^{n-1} \left(1 - \frac{n}{s}\right)^{-n} \left(1 - \frac{n}{s}\right)^{s} \frac{1}{s}\,. && (2.12)
\end{aligned}$$

Because $((s-1)/s)^{n-1}(1 - n/s)^{-n}(1 - n/s)^s \to 1 \cdot 1 \cdot e^{-n}$ when $s \to \infty$, the terms in the sum (2.12) approach the terms of a harmonic series multiplied by a constant, and thus $C(M, \mathbb{N}_+^n) = \infty$. A direct consequence is that for all probability mass functions $f : \mathbb{N}_+^n \to [0,1]$ the regret $\mathrm{REG}_M(f, \mathbf{x})$ is unbounded in $\mathbb{N}_+^n$.

2.5.1 Restricting the parameter range

A potential solution is to modify the model class so that all ML parameter estimates are not possible. Let $\theta_0 \in \,]0,1[$. The parametric complexity of the model class $M_{\theta_0} = \{p(\cdot;\, \theta) \mid \theta \in [\theta_0, 1]\}$ is finite, since

$$C(M_{\theta_0}, \mathbb{N}_+^n) = \sum_{s=n}^{\lfloor n/\theta_0 \rfloor} \sum_{\substack{\mathbf{x} \in \mathbb{N}_+^n, \\ \sum_{i=1}^n x_i = s}} \left(1 - \frac{n}{s}\right)^{s-n} \left(\frac{n}{s}\right)^n \tag{2.13}$$

$$+ \sum_{s=\lfloor n/\theta_0 \rfloor + 1}^{\infty} \sum_{\substack{\mathbf{x} \in \mathbb{N}_+^n, \\ \sum_{i=1}^n x_i = s}} (1-\theta_0)^{s-n} \theta_0^n \,, \tag{2.14}$$

$$= 1 + \sum_{s=n}^{\lfloor n/\theta_0 \rfloor} \binom{s-1}{n-1} \left(\left(1 - \frac{n}{s}\right)^{s-n} \left(\frac{n}{s}\right)^n - (1-\theta_0)^{s-n} \theta_0^n \right) .$$

We can hence encode every sequence in $\mathbb{N}_+^n$ with the corresponding NML distribution $p^{\mathrm{NML}}_{M_{\theta_0}, \mathbb{N}_+^n}$. But even if $\mathrm{REG}_{M_{\theta_0}}(p^{\mathrm{NML}}_{M_{\theta_0}}, \mathbf{x})$ is constant for all $\mathbf{x} \in \mathbb{N}_+^n$, the meaningfulness of the model class M_{θ_0} depends on the relationship between the parameter θ_0 and the data. The regret with regard to the original model class M is plotted in Figure 2.1. By making θ_0 smaller we can increase the size of the set in which this regret is constant. In the same time, the code lengths for sequences with a large ML parameter estimate increase. In practice, we should have some prior knowledge about the data in order to be able to choose θ_0 so that the event of seeing a sequence $\mathbf{x} \in \mathbb{N}_+^n$ with $\hat{\theta}_M(\mathbf{x}) < \theta_0$ becomes unlikely enough.

2.5.2 Restricting the data

An alternative to restricting the model class is to restrict the data. In this case, we might consider instead of $\mathbb{N}_+^n$ the set $D = \{\mathbf{x} \in \mathbb{N}_+^n \mid \hat{\theta}_M(\mathbf{x}) \geq \theta_0\}$ where $\theta_0 \in \,]0,1[$. Taking advance of the calculations in (2.13)–(2.14), in which the second summation goes over probabilities over a single distribution, we see that $0 < C(M_{\theta_0}, \mathbb{N}_+^n) - C(M, D) < 1$. In addition, because $C(M, D) \geq 1$, the code length difference $-\log_2 p^{\mathrm{NML}}_{M_{\theta_0}, \mathbb{N}_+^n}(\mathbf{x}) - (-\log_2 p^{\mathrm{NML}}_{M,D}(\mathbf{x})) \leq 1$, when $\mathbf{x} \in D$. In other words, the encoding of sequences in D is only slightly more

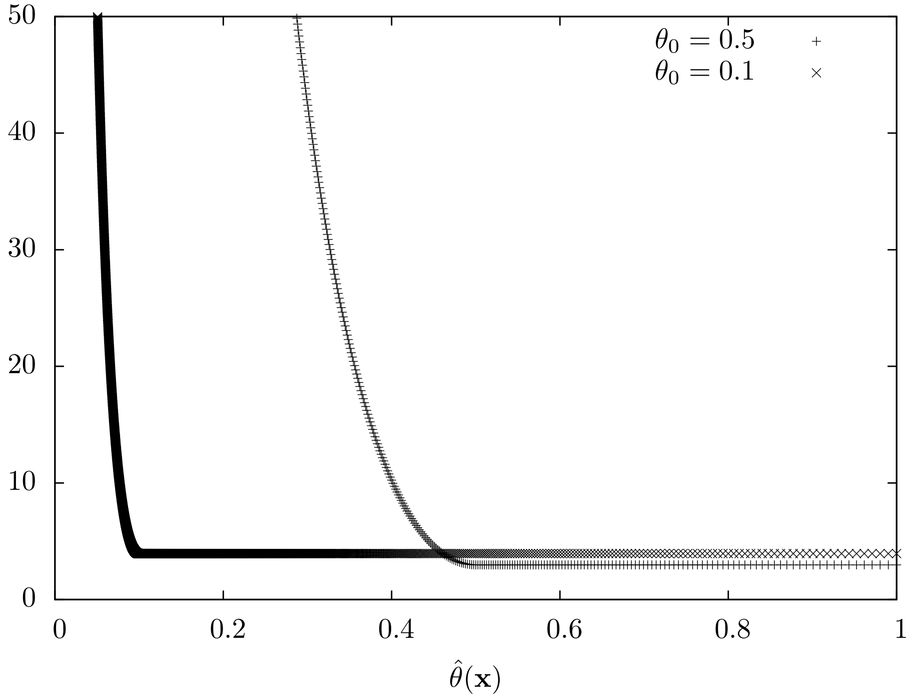


Figure 2.1: The regret $\mathrm{REG}_M(p^{\mathrm{NML}}_{M_{\theta_0},\mathbb{N}^n_+}, \mathbf{x})$ as a function of $\hat{\theta}_M(\mathbf{x})$ for two different values of θ_0 when $n = 100$.

effective using $p^{\mathrm{NML}}_{M,D}$ than with $p^{\mathrm{NML}}_{M_{\theta_0},\mathbb{N}^n_+}$. Encoding of elements in $\mathbb{N}^n_+ \setminus D$ is of course impossible with $p^{\mathrm{NML}}_{M,D}$. One might ask, whether the model class M is a good choice when we assume that the data belong to D, as the domain of the probabilities in M is whole $\mathbb{N}^n_+$. But apart from the philosofical aspect, a model class in which the distributions would be normalized for the domain D would be considerably more difficult to handle in calculations than M.

For our example, we shall consider yet another and possibly more natural way to restrict the data. Imagine that a single element in the data sequence comes from a test in which the number of Bernoulli trials needed to get the first "success" is recorded. It is then reasonable to assume that before the tests an upper limit $m - 1 \geq 1$ was set for the number of trials in a single test, and if no success was seen in the first $m - 1$ trials, the value of the test was recorded as m. That is, we limit the domain of the data through single elements of the sequence, not through the sum of the sequence's elements. Interpreting value m as an indicator of the event "success did not occur before the trial number m", the probability mass function for a single outcome is $p_1 : \{1, 2, \ldots, m\} \to [0, 1]$,

$$p_1(x;\, \theta) = \begin{cases} (1-\theta)^{x-1}\theta & \text{if } x \in \{1, 2, \ldots, m-1\} \\ (1-\theta)^{m-1} & \text{if } x = m\,. \end{cases}$$

Then the probability of sequence $\mathbf{x} \in \{1, 2, \ldots, m\}^n$ is

$$\begin{aligned} p_n(\mathbf{x};\, \theta) &= \left((1-\theta)^{m-1}\right)^k \cdot (1-\theta)^{s-km-(n-k)} \theta^{n-k} \\ &= (1-\theta)^{s-n}\, \theta^{n-k} \end{aligned}$$

where $k = \sum_{i=1}^n \mathbf{1}(x_i = m)$ and $s = \sum_{i=1}^n x_i$. Let the model class be $M' = \{p_n(\cdot;\, \theta) \mid \theta \in [0, 1]\}$. Simple calculations yield that $\hat{\theta}_{M'}(\mathbf{x}) = (n-k)/(s-k)$ where k and s are defined as functions of $\mathbf{x}$ similarly as above.

The normalizing term is thus

$$\begin{aligned} &C(M', \{1, 2, \ldots, m\}^n) \\ &\quad = \sum_{\mathbf{x} \in \{1,\ldots,m\}^n} p_n(\mathbf{x};\, \hat{\theta}_{M'}(\mathbf{x})) \\ &\quad = \sum_{k=0}^{n} \binom{n}{k} \sum_{t=n-k}^{(n-k)(m-1)} \left[\sum_{j=0}^{n-k} (-1)^j \binom{n-k}{j} \binom{t - j(m-1) - 1}{n-k-1} \right] \\ &\qquad\qquad \cdot \left(1 - \frac{n-k}{t+km-k}\right)^{t+km-n} \left(\frac{n-k}{t+km-k}\right)^{n-k} . \end{aligned}$$

Again, k denotes the number of occurrences of m in sequence $\mathbf{x}$. The second summation goes over all possible values for $t = \sum_{i=1}^{n} x_i \mathbf{1}(x_i < m) = \sum_{i=1}^{n} x_i - km$. In the brackets is the size of the set $\{(y_1, y_2, \ldots, y_{n-k}) \in \{1, 2, \ldots, m-1\}^{n-k} \mid \sum_{i=1}^{n-k} y_i = t\}$, in other words, it is the number of compositions of t into $n-k$ terms such that each term belongs to the set $\{1, 2, \ldots, m-1\}$ [1, Equation (3.2 E)]. For convenience, we use above the notational convention $\binom{0}{0} = \binom{-1}{-1} = 1$.

In general, if there is only very little prior information about the domain of the data, arbitrarily restricting the domain or the model class are hardly reasonable solutions to the problem of infinite parametric complexity. In the next section we mention more suitable alternatives.

2.6 Infinite Parametric Complexity and a Variable Regret

In the last section we saw how the problem of infinite parametric complexity can be circumvented by restricting either the parameter space directly or by restricting the domain of the data. Both methods require some prior information of the data. Too broad bounds for the parameters, or for the data, lead to large parametric complexities of model classes, which may be a problem in practice. Let us consider the clustering scheme of Paper III as an example. If we use NML code for the encoding of the data given the clustering, the code length of every cluster subsequence includes a constant that depends only on the bounds. For all data sets, it is possible to make that constant so large that the total code length is minimized by putting all data elements into one cluster.

The need to avoid pitfalls of arbitrary assumptions has recently led to more flexible encoding schemes for model classes with an infinite parametric complexity. Using the terminology of Grünwald, important examples are meta-two-part coding, renormalized maximum likelihood, NML with luckiness, and conditional NML [11, Chapter 11]. Next we describe the four methods briefly, but only NML with luckiness is relevant for this thesis. We do not cover sequential methods like the sequentially normalized maximum likelihood (SNML) [36, Chapter 9][13] here.

The *meta-two-part coding* [32] is a simplistic method. There we carve the parameter space to pieces that correspond to model classes with well-defined NML distributions and decide an encoding scheme for them. The code length is the minimized sum of the code for a model class M_i and the NML code length of the data according to M_i. As with any two-part code, it is easy to show that it is usually not the shortest code, because it is typically possible to encode the data according to two different model classes. But probably the most difficult problem is to decide how to carve up the parameter space in a sensible way.

Renormalized maximum likelihood (RNML) [33] introduces hyperparameters that bound the parameter space and which are then treated like normal parameters in the NML calculations. The idea is to find after possibly several renormalizations a code length function in which the hyperparameters do not affect the model selection task any more. However, the strategy does not work well with all model selection problems.

NML with luckiness tries to achieve an acceptable regret that is a function of the ML parameters only. The main difference to the previous methods is that NML with luckiness concentrates directly on the regret and pays less attention to the encoding strategies as such. The word "luckiness" refers to fact that using an adequate coding, the regret is not too large with most data, and if we are lucky, we can get an even shorter regret. We discuss details of this method in Section 3.2. The code lengths for various model classes in Papers II and III fall mostly in the NML with luckiness category. There exists also a variant of NML with luckiness, in which one replaces the original model class $M = \{f(\cdot;\, \theta) \mid \theta \in \Theta\}$ with $M' = \{q(\theta) f(\cdot;\, \theta) \mid \theta \in \Theta\}$, where q is a density of the parameter vector [16]. This is a departure from the basic idea of MDL, as we were originally interested in minimizing the worst-case regret with regard to M, not M'.

Conditional NML refers in Grünwald's terminology to a technique where a part of the data, perhaps just a few first points, are not encoded in an optimal way – or they are assumed to be known by the receiver. After the initial data, the rest can be encoded using NML.

Chapter 3

Codes for Model Classes

We may encounter at least two kinds of problems when designing an NML based code for a model class. Sometimes the main difficulty is to calculate the code length efficiently enough so that it can be used in practical applications. Section 3.1 covers encoding of a clustering sequence, which is a good example of this kind of challenge: the parametric complexity of a multinomial model class with restricted data can be calculated using a recurrence relation, and finding the relation is best done using generating functions as a tool.

In Section 3.2, we discuss the infinite parametric complexity case, in which the very meaning of “a good code” is somewhat ambiguous. A code minimizing the worst-case regret simply does not exist, and we must find a satisfying compromise. Our approach can be categorized as NML with luckiness, as we concentrate on the aspect of how the regret behaves as a function of the maximum likelihood parameter estimates.

3.1 Encoding of a Clustering Sequence

In Paper IV we give an effective way to calculate the NML for a model class with multinomial distributions in a situation when the ML parameters are known to be positive. The case is relevant for clustering, and the corresponding NML is a more natural choice than the general multinomial NML in applications like those in Paper III. In this section, we give some background and outline the code length calculations. The interested reader is advised to consult Paper IV for technical details.

A natural presentation for a clustering of a data sequence is a sequence of labels, in which the actual label values are irrelevant. The only information we are interested in is which elements of the sequence have the same labels. In this context, (⚡, ⚡, ⛅, ☼, ☼) and $(1, 1, -10, 314, 314)$ are just different presentations of the same information. From now on, we call such a label sequence a *clustering sequence.*

When choosing model classes and families, we aim to capture regularities of the data. If our code uses many bits for describing something that we consider simple, we have perhaps not chosen an appropriate model family, or some reason necessitates the use of simplified models. Let us illustrate the situation with a bit vector example. Let $M = \{p(\cdot;\theta) \mid \theta \in [0,1]\}$ where $p : \{0,1\}^n \to [0,1]$ is the probability mass function of a sequence of n independently and identically-distributed Bernoulli variables. The Kolmogorov complexity of the sequence $S = 010101\ldots010101$ is small, but if we define its complexity by means of the model class M, sequence S is maximally random.

In our encoding scheme for clustering sequences we assume that the elements are independent and identically-distributed. Although such models do not capture patterns where the order of elements is relevant, they are widely used because of their simplicity. One should also notice that the independence assumption applies only to cluster labels: the cluster data subsequences may still be modelled e.g. using Markov chains.

A clustering sequence represents a partition of the data. We point out, that the use of distributions on partitions is an important advance in non-parametric Bayesian modelling (see e.g. [27]), and the so called Chinese Restaurant Process with two parameters is the best known method in that area. The corresponding distributions could be used in an NML based approach too, if we are able to find the ML estimates for the parameters [6]. However, as we shall see later, our approach has a characteristic that matches the MDL philosophy well: we consider all choices for the number of clusters equally probable.

We take one of the simplest ways to model clustering sequences, using model classes with multinomial distributions. Let us consider one example, in which we write the length of the sequence as n and the number of distinct labels as k. If we know that $k = 1$, there is only one possible sequence (which has of course different representations). And on the condition that $k = n$, there is also no uncertainty about the sequence. Therefore, it is natural to require from the coding strategy that the clustering sequences $(1, 1, \ldots, 1)$ and $(1, 2, \ldots, n)$ have short code lengths when the number of distinctive labels is known. But if we use the general multinomial NML (e.g. [17]), the code length is short only in the first case, because the parametric complexity of the model classes grows monotonically as a function of k.

Before inspecting multinomial NML, we start with some formal definitions. We can assume without loss of generality that in a clustering sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ it holds that $x_1 = 1$ and if $x_i \neq x_j$ for all $j \in \{1, 2, \ldots, i-1\}$, then $x_i = 1 + \max\{x_j \mid j < i\}$. Let $D \subset \{1, 2, \ldots, k_0\}^n$ denote the set consisting of all such n-sequences with at most k_0 different labels, and let $D_k = \{\mathbf{x} \in D \mid \max(\mathbf{x}) = k\}$ for $k \in \{1, 2, \ldots, k_0\}$. Let the parameter space for multinomial distributions with k parameters be $\Delta_{k-1} = \{(p_1, p_2, \ldots, p_k) \in [0,1]^k \mid \sum_{i=1}^k p_i = 1\}$, and let $\mathbf{p} = (p_1, p_2, \ldots, p_k) \in \Delta_{k-1}$. The probability of sequence $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \{1, 2, \ldots\}^n$ is $P_k(\mathbf{y}; \mathbf{p}) = \prod_{i=1}^n p_{y_i}$ if $\mathbf{y} \in \{1, 2, \ldots, k\}^n$, and $P_k(\mathbf{y};\, \mathbf{p}) = 0$ otherwise.

For all $k \in \{1, 2, \ldots, k_0\}$, let

$$M_k = \{P_k(\cdot;\, \mathbf{p}) \mid \mathbf{p} \in \Delta_{k-1}\} \tag{3.1}$$

be a model class. The maximum likelihood value according to M_k is for $\mathbf{x} \in D_k$

$$P_k(\mathbf{x};\, \hat{\mathbf{p}}_k(\mathbf{x})) = \prod_{i=1}^{k} \left(\frac{n_i(\mathbf{x})}{n} \right)^{n_i(\mathbf{x})} \tag{3.2}$$

where $n_i(\mathbf{x})$ is the number of occurrences of the element i in $\mathbf{x}$, and $\hat{\mathbf{p}}_k(\mathbf{x}) = (n_1(\mathbf{x}), n_2(\mathbf{x}), \ldots, n_k(\mathbf{x}))/n$ is the ML parameter estimator (we use again the convention $0^0 \equiv 1$). It would have been possible to scale P_k so that the resulting probability mass function sums up to 1 over the set D_k. However, we prefer the simpler notation because the scaling would have had no effect on the NML distribution.

At this point, it is useful to briefly compare the modelling problem at hand to a more typical instance. Consider the model class selection problem with normal mixture models in Section 2.1 for comparison. In that case every possible data sequence has a positive ML density in all the model classes, and the ML density of a data sequence tends to increase when we change to a model class with more components. Let then $\mathbf{x} \in D_m$ be a clustering sequence, and let $\{M_1, M_2, \ldots, M_{k_0}\}$ be a collection of model classes defined as in (3.1). The maximum likelihood of $\mathbf{x}$ is 0 in the model classes $M_1, M_2, \ldots, M_{m-1}$. Moreover, the ML of $\mathbf{x}$ is equal according to all the model classes $M_m, M_{m+1}, \ldots, M_{k_0}$. It is awkward to consider $\{M_1, M_2, \ldots, M_{k_0}\}$ as a model family, since the supports of the distributions in the model classes are not equal. The usual overfitting problem does not exist, and the choice of the "best" model class is trivial. That is why we first divide the problem into smaller parts and study the encoding of elements in D_i according to the model class M_i. Then we put the distributions with separate domains together to achieve a distribution over D.

For $\mathbf{x} \in D_k$, where $k \in \{1, 2, \ldots, k_0\}$, the NML distribution is $P^{\text{NML}}_{M_k, D_k}$:

$D_k \to [0,1]$,

$$P^{\mathrm{NML}}_{M_k,D_k}(\mathbf{x}) = \frac{P_k(\mathbf{x};\, \hat{\mathbf{p}}_k(\mathbf{x}))}{C(M_k, D_k)} = \frac{P_k(\mathbf{x};\, \hat{\mathbf{p}}_k(\mathbf{x}))}{\sum_{\mathbf{y}\in D_k} P_k(\mathbf{y};\, \hat{\mathbf{p}}_k(\mathbf{y}))} = \frac{P_k(\mathbf{x};\, \hat{\mathbf{p}}_k(\mathbf{x}))}{(1/k!)\sum_{\mathbf{y}\in D'_k} P_k(\mathbf{y};\, \hat{\mathbf{p}}_k(\mathbf{y}))}\,. \tag{3.3}$$

where $D'_k = \{\mathbf{y} \in \{1,2,\ldots,k\}^n \mid n_1(\mathbf{y}),\ldots,n_k(\mathbf{y}) \geq 1\}$. Paper IV represents a recurrence relation, with which the normalization factor $C(M_k, D_k)$ can be calculated efficiently. Using the notation of the paper, we write

$$\mathcal{C}_1(k,n) = k!\, C(M_k, D_k) = \sum_{\substack{\mathbf{y}\in\{1,2,\ldots,k\}^n,\\ n_1(\mathbf{y}),\ldots,n_k(\mathbf{y})\geq 1}} P_k(\mathbf{y};\, \hat{\mathbf{p}}_k(\mathbf{y}))\,. \tag{3.4}$$

Let $n \in \{3,4,\ldots,\}$. It holds for all $k \in \{1,2,\ldots,n-2\}$ (see [26] and Paper IV) that

$$\mathcal{C}_1(k+2,n) + 2\mathcal{C}_1(k+1,n) = \left(\frac{n}{k} - 1\right)\mathcal{C}_1(k,n)\,. \tag{3.5}$$

With recurrence (3.5) we can calculate $C(M_k, D_k)$ in $O(n)$ time.

Inside each D_k, the NML distribution gives now high probabilities for intuitively simple sequences. For example, $P^{\mathrm{NML}}_{M_n,D_n}((1,2,\ldots,n)) = 1$. The separate NML distributions can be combined by normalization, and writing $m(\mathbf{x}) = \max(\mathbf{x})$, we obtain $P : D \to [0,1]$,

$$P(\mathbf{x}) = \frac{P^{\mathrm{NML}}_{M_{m(\mathbf{x})},\, D_{m(\mathbf{x})}}(\mathbf{x})}{\sum_{\mathbf{y}\in D} P^{\mathrm{NML}}_{M_{m(\mathbf{y})},\, D_{m(\mathbf{y})}}(\mathbf{y})} = \frac{P^{\mathrm{NML}}_{M_{m(\mathbf{x})},\, D_{m(\mathbf{x})}}(\mathbf{x})}{\sum_{i=1}^{k_0}\sum_{\mathbf{y}\in D_i} P^{\mathrm{NML}}_{M_i,\, D_i}(\mathbf{y})} = \frac{1}{k_0}\, P^{\mathrm{NML}}_{M_{m(\mathbf{x})},\, D_{m(\mathbf{x})}}(\mathbf{x})\,. \tag{3.6}$$

The result is thus an NML distribution according to the model class $\{P^{\mathrm{NML}}_{M_i,D_i} \mid i \in \{1,2,\ldots,k_0\}\}$ and the domain D.

In Paper IV we prove that $\mathcal{C}_1(k,n)$ is maximized with a fixed n when $k = \lfloor n/4 \rfloor + 1$ or $k = \lceil n/4 \rceil + 1$ (Figure 3.1). Figure 3.2 illustrates the

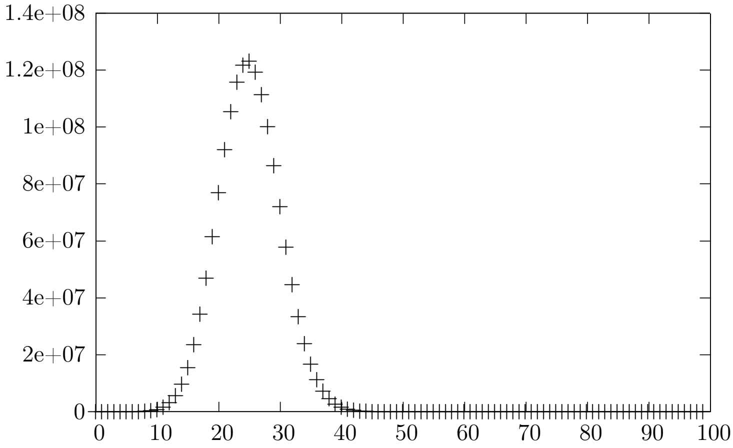


Figure 3.1: $\mathcal{C}_1(k,n)$ from (3.4) as a function of k when $n = 100$.

difference between the parametric complexities of the general multinomial and the constrained model class. The normalizing sum is in the general case

$$\mathcal{C}_0(k,n) = \sum_{\substack{\mathbf{y}\in\{1,2,\ldots,k\}^n,\\ n_1(\mathbf{y}),\ldots,n_k(\mathbf{y})\geq 0}} P_k(\mathbf{y};\, \hat{\mathbf{p}}_k(\mathbf{x}))\,. \tag{3.7}$$

In classical clustering applications, it is assumed that the number of clusters is a small constant compared to the sample size. Then, the difference $\log \mathcal{C}_0(k,n) - \log \mathcal{C}_1(k,n)$ is small, and the choice of the multinomial model class has probably little effect on the clustering method. But if the number of clusters is a growing function of the sample size – new components arise when the time passes, but each component produces only a constant number of points – the difference can be significant.

3.2 Model Classes with Infinite Parametric Complexity

In Papers I and II we present codes for uniform and Gaussian model classes, including some multidimensional cases, and we propose practical solutions to the problem of infinite parametric complexity in these cases. Paper III introduces also codes for shifted exponential, Laplace and shifted half-normal distributions in one dimension. In this section, we take the perspective of Paper III and discuss what kind of requirements a code (or the corresponding

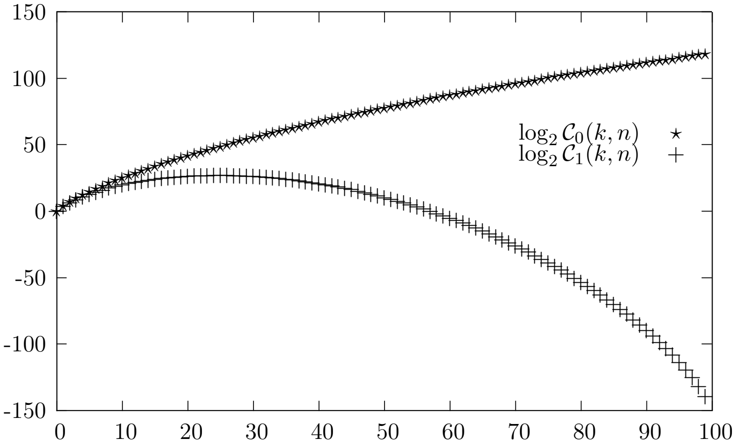


Figure 3.2: The parametric complexities for two types of multinomial model classes with $n = 100$. The sums $\mathcal{C}_0(k,n)$ and $\mathcal{C}_1(k,n)$ are defined in (3.7) and (3.4), respectively.

density) should fill if the NML distribution does not exist, and how our codes fulfil these requirements.

All the distributions that we cover in this section belong to the location-scale family of distributions (see Table 3.1, page 27). In the one-dimensional case, the density function of such a distribution can be written in the form $f(\cdot;\, \alpha, \beta) : \mathbb{R} \to [0, \infty[$, where $\alpha \in \mathbb{R}$ is the location parameter and $\beta \in \,]0, \infty[$ the scale parameter. Density $f(\cdot;\, \alpha, \beta)$ can be shifted in the sense that $f(x;\, \alpha, \beta) = f(x + \Delta;\, \alpha + \Delta, \beta)$ for all $x, \Delta \in \mathbb{R}$. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ be a data sequence, and let $\hat{\alpha}(\mathbf{x})$ and $\hat{\beta}(\mathbf{x})$ be the maximum likelihood parameter estimates for $\mathbf{x}$. Assume that the elements of a sequence are independent and identically-distributed. With all the distributions in Table 3.1, the maximum likelihood value of a sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \{\mathbf{y} \in \mathbb{R}^n \mid \min(\mathbf{y}) \neq \max(\mathbf{y})\}$ depends only on the ML scale parameter in the following sense: $f_n(\mathbf{x};\, \hat{\alpha}(\mathbf{x}), \hat{\beta}(\mathbf{x})) = g^{\mathrm{ML}}(\hat{\beta}(\mathbf{x}))$, where $f_n(\mathbf{x};\, \alpha, \beta) = \prod_{i=1}^{n} f(x_i;\, \alpha, \beta)$ and $g^{\mathrm{ML}} : \,]0, \infty[\, \to [0, \infty[$.

Let T denote the model class type (uniform, exponential etc.), and let the model class of type T be $M_T = \{f_T(\cdot;\, \alpha, \beta) \mid \alpha \in \mathbb{R},\, \beta \in \,]0, \infty[\,\}$. When the data domain is $D = \{\mathbf{x} \in \mathbb{R}^n \mid \hat{\alpha}_T(\mathbf{x}) \in [\alpha_1, \alpha_2],\ \hat{\beta}_T(\mathbf{x}) \in [\beta_1, \beta_2]\}$, where $\alpha_1 < \alpha_2$ and $0 < \beta_1 < \beta_2$, the normalizing factor

$$C(M_T, D) = \int_D f_T(\mathbf{x};\, \hat{\alpha}_T(\mathbf{x}), \hat{\beta}_T(\mathbf{x}))$$

is finite and it can be calculated in a straightforward way (Table 3.4). The corresponding NML density serves as a technical starting point for the code derivations in the case $D = \mathbb{R}^n$ in Papers I, II and III. We pass the details but discuss the intuitive properties that density $\tilde{f}$ corresponding to model class M_T and data domain $\mathbb{R}^n$ should have.

The obvious requirement is that $\tilde{f}$ is positive everywhere in $\mathbb{R}^n$. It is also natural to assume that $\tilde{f}$ is continuous, and that it is a function of $\hat{\alpha}(\mathbf{x})$ and $\hat{\beta}(\mathbf{x})$ only, in other words, that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ the identity $(\hat{\alpha}_T(\mathbf{x}), \hat{\beta}_T(\mathbf{x})) = (\hat{\alpha}_T(\mathbf{y}), \hat{\beta}_T(\mathbf{y}))$ implies $\tilde{f}(\mathbf{x}) = \tilde{f}(\mathbf{y})$. The maximum likelihood densities of the distributions we consider are proportional to $\hat{\beta}_T(\mathbf{x})^{-n}$, and they are thus not defined when $\hat{\beta}(\mathbf{x}) = 0$ (Tables 3.2 and 3.3). Because of the requirement that $\tilde{f}$ should be continuous and positive everywhere, it is inevitable that regret $\mathrm{REG}_{M_T}(\tilde{f}, \mathbf{x})$ grows unbounded as $\hat{\beta}(\mathbf{x})$ approaches 0 from above. In practice, the data are given with some fixed maximum precision. Therefore it is natural to allow the regret to be unbounded in the set $\{\mathbf{x} \in \mathbb{R}^n \mid \hat{\beta}_T(\mathbf{x}) < \epsilon\}$, where $\epsilon > 0$ is a constant. But even then, the regret has to grow as a function of $|\hat{\alpha}_T(\mathbf{x})|$, so that $\tilde{f}$ would integrate to unity over $\mathbb{R}^n$. When a data sequence $\mathbf{x}$ is shifted away from the origin, $|\hat{\alpha}(\mathbf{x})|$ grows and $g^{\mathrm{ML}}(\hat{\beta}(\mathbf{x}))$ remains unchanged. In this context, a condition that is conceptually somewhat similar to minimizing the worst-case regret, is to require that the regret grows asymptotically slowly as a function of $|\hat{\alpha}(\mathbf{x})|$.

In Paper III we derive densities that have the form

$$\tilde{f}(\mathbf{x}) = \begin{cases} g^{\mathrm{ML}}(\hat{\beta}(\mathbf{x}))\, d_n\, \epsilon\, p(\hat{\alpha}(\mathbf{x})) & \text{if } \hat{\beta}(\mathbf{x}) \geq \epsilon \\ g^{\mathrm{ML}}(\epsilon)\, d_n\, \epsilon\, p(\hat{\alpha}(\mathbf{x})) & \text{if } 0 \leq \hat{\beta}(\mathbf{x}) < \epsilon\,. \end{cases}$$

The coefficient d_n depends on the model class (see Table 3.5), and $p : \mathbb{R} \to \mathbb{R}_+$ is a continuous density function that gets positive values everywhere. As before, $g^{\mathrm{ML}}(\hat{\beta}(\mathbf{x})) = f(\mathbf{x};\, \hat{\alpha}(\mathbf{x}), \hat{\beta}(\mathbf{x}))$ denotes the maximum likelihood value in the model class. In all the cases we consider, $g^{\mathrm{ML}} \circ \hat{\beta}$ is a continuous function.

The simple requirements for $\tilde{f}$ – positivity, continuity and dependence on $\mathbf{x}$ only through $\hat{\alpha}(\mathbf{x})$ and $\hat{\beta}(\mathbf{x})$ – are clearly fulfilled. When $\hat{\beta}(\mathbf{x}) \geq \epsilon$, the regret is $-\log(d_n \epsilon) - \log p(\hat{\alpha}(\mathbf{x}))$. We have parameterized p with parameter $a \in \mathbb{R}_+$ in Paper III so that density $p(\hat{\alpha}(\mathbf{x}))$ is constant when $\hat{\alpha}(\mathbf{x}) \in [-a, a]$. Thus the regret is constant in set $\{\mathbf{x} \in \mathbb{R}^n \mid \hat{\alpha}(\mathbf{x}) \in [-a, a],\, \hat{\beta}(\mathbf{x}) \geq \epsilon\}$. Because our p, which is based on Rissanen's prior for integers [30], has longer tails than any commonly used density function, the regret grows asymptotically very slowly as a function of $|\hat{\alpha}(\mathbf{x})|$. The amount of probability mass in the tails of p can be further controlled with another parameter.

In Paper II, we have derived a density that resembles $\tilde{f}$ but depends on a technical prior for the scale parameter as well as for the location parameter. From the perspective of the MDL principle this seems somewhat problematic, as the regret changes when either $\hat{\alpha}(\mathbf{x})$ or $\hat{\beta}(\mathbf{x})$ change. Moreover, when $\hat{\beta}(\mathbf{x}) \geq \epsilon$, the contribution of $\hat{\beta}(\mathbf{x})$ to the regret is e.g. in the case of the uniform model class $-2 \log \hat{\beta}(\mathbf{x}) - \log p_{\beta}(\hat{\beta}(\mathbf{x}))$. Even if it is impossible to find a code for a model class with an infinite parametric complexity without assuming something about the data, it is in our opinion advisable to keep the assumptions as simple as possible. The cleaner approach of Paper III requires that we should choose only an effective lower bound ϵ for $\hat{\beta}(\mathbf{x})$, and a density p that determines how much we favour data centered in the vicinity of the origin.

Distribution	**Density** $f_T(x;\, \alpha, \beta)$	**Support**
Uniform	$1/2\beta$	$[\alpha-\beta, \alpha+\beta]$
Exponential	$1/\beta \cdot \exp(-(x-\alpha)/\beta)$	$[\alpha, \infty[$
Mirrored exponential	$1/\beta \cdot \exp(-(\alpha-x)/\beta)$	$]-\infty, \alpha]$
Laplace	$1/2\beta \cdot \exp(-\lvert x-\alpha\rvert/\beta)$	$\mathbb{R}$
Normal	$1/\sqrt{2\pi\beta^2} \cdot \exp(-(x-\alpha)^2/2\beta^2)$	$\mathbb{R}$
Half-normal	$\sqrt{2/\pi\beta^2} \cdot \exp(-(x-\alpha)^2/2\beta^2)$	$[\alpha, \infty[$
Mirrored half-normal	$\sqrt{2/\pi\beta^2} \cdot \exp(-(x-\alpha)^2/2\beta^2)$	$]-\infty, \alpha]$

Table 3.1: One-dimensional distributions of the location-scale family.

Distribution	**ML estimate** $\hat{\beta}(\mathbf{x})$	**ML density**
		$f_T(\mathbf{x}; \hat{\alpha}(\mathbf{x}), \hat{\beta}(\mathbf{x}))$
Uniform	$(\max(\mathbf{x}) - \min(\mathbf{x}))/2$	$1/(2\hat{\beta}(\mathbf{x}))^n$
Exponential	$(1/n)\sum_{i=1}^n (x_i - \min(\mathbf{x}))$	$1/(e\hat{\beta}(\mathbf{x}))^n$
Mirrored exponential	$(1/n)\sum_{i=1}^n (\max(\mathbf{x}) - x_i)$	$1/(e\hat{\beta}(\mathbf{x}))^n$
Laplace	$(1/n)\sum_{i=1}^n \lvert x_i - x_{\lceil n/2 \rceil}\rvert$	$1/(2e\hat{\beta}(\mathbf{x})^n$

Table 3.2: Maximum likelihood scale-parameter estimates and ML densities for a sequence with n identically and independently-distributed elements.

Distribution	**ML estimate** $\hat{\beta}^2(\mathbf{x})$	**ML density**
		$f_T(\mathbf{x}; \hat{\alpha}(\mathbf{x}), \hat{\beta}(\mathbf{x}))$
Normal	$(1/n)\sum_{i=1}^n (x_i - (1/n)\sum_{i=1}^n x_i)^2$	$1/(2\pi e\hat{\beta}^2(\mathbf{x}))^{n/2}$
Half-normal	$(1/n)\sum_{i=1}^n (x_i - \min(\mathbf{x}))^2$	$(2/\pi e\hat{\beta}^2(\mathbf{x}))^{n/2}$
Mirrored half-normal	$(1/n)\sum_{i=1}^n (x_i - \max(\mathbf{x}))^2$	$(2/\pi e\hat{\beta}^2(\mathbf{x}))^{n/2}$

Table 3.3: Maximum likelihood estimates for squared scale-parameter estimates and ML densities for a sequence with n identically and independently-distributed elements.

Model class type	**Normalizing factor** / $(\alpha_2 - \alpha_1)(\beta_1^{-1} - \beta_2^{-1})$
Uniform	$\dfrac{n(n-1)}{2}$
(Mirrored) Exponential	$\dfrac{1}{(n-2)!}\left(\dfrac{n}{e}\right)^n$
Laplace	$\dfrac{n-1}{(\lceil n/2\rceil - 1)!\,(n - \lceil n/2\rceil)!}\left(\dfrac{n}{2e}\right)^n$
Normal	$\dfrac{2}{\sqrt{\pi}\,\Gamma((n-1)/2)}\left(\dfrac{n}{2e}\right)^{n/2}$
(Mirrored) Half-normal	$\dfrac{4\sqrt{n}}{\sqrt{\pi}\,\Gamma((n-1)/2)}\left(\dfrac{n}{2e}\right)^{n/2}$

Table 3.4: Normalizing factors divided by $(\alpha_2 - \alpha_1)(\beta_1^{-1} - \beta_2^{-1})$ in the NML density when the domain of the data is $D = \{\mathbf{x} \in \mathbb{R}^n \mid \hat{\alpha}(\mathbf{x}) \in [\alpha_1, \alpha_2],\ \hat{\beta}(\mathbf{x}) \in [\beta_1, \beta_2]\}$.

Model class type	**Coefficient** d_n
Uniform	$\dfrac{2}{n^2}$
(Mirrored) Exponential	$\dfrac{e^n(n-1)!}{n^{n+1}}$
Laplace	$(2e)^n\dfrac{(\lceil n/2\rceil - 1)!(n - \lceil n/2\rceil)!}{n^{n+1}}$
Normal	$\dfrac{\sqrt{\pi}}{2}\left(\dfrac{2e}{n}\right)^{n/2}\dfrac{n-1}{n}\,\Gamma\left(\dfrac{n-1}{2}\right)$
(Mirrored) Half-normal	$\dfrac{\sqrt{\pi}}{4}\left(\dfrac{2e}{n}\right)^{n/2}\dfrac{n-1}{n^{3/2}}\,\Gamma\left(\dfrac{n-1}{2}\right)$

Table 3.5: Coefficient d_n in $\tilde{f}(\mathbf{x}) = g^{\mathrm{ML}}(\hat{\beta}(\mathbf{x}))d_n\epsilon\, p(\hat{\alpha}(\mathbf{x}))$ for $\mathbf{x} \in \{\mathbf{y} \in \mathbb{R}^n \mid \hat{\beta}(\mathbf{y}) \geq \epsilon\}$.

Chapter 4

Applications

The number of existing practical applications based on the NML is relatively small compared to applications using some other form of the MDL. Especially taking account the important position of the NML in the current MDL research, the reasons are presumably mostly related to the problems we have discussed earlier in this thesis. To name some examples of the practical use of the NML, there are applications in bioinformatics [20, 21, 22, 46, 47], data clustering [19] and wavelet denoising [33, 39, 38].

In this chapter we introduce some applications of the NML distributions derived in the previous chapter. A common theme is clustering, either in one or multidimensional spaces. The cluster assignments are always binary – that is, every point belongs to exactly one cluster – which simplifies the scheme significantly. We choose the number of clusters, the clustering itself and the types of the clusters by maximizing the density

$$f(k, \mathbf{c}, \mathbf{t}, \mathbf{x}) = \frac{1}{K} \, p_k^{\mathrm{NML}}(\mathbf{c}) \, \frac{1}{T^k} \prod_{i=1}^{k} f_{t_i}(\mathbf{y}_i) \tag{4.1}$$

where $k \in \{1, 2, \dots, K\}$ is the number of clusters, $\mathbf{c} \in \{1, 2, \dots, k\}^n$ is the clustering sequence, $\mathbf{t} = (t_1, t_2, \dots, t_k) \in \{1, 2, \dots, T\}^k$ are the types of the clusters, $\mathbf{x} \in (\mathbb{R}^d)^n$ is the data sequence, and $\mathbf{y}_i$ is the subsequence corresponding to cluster number i. We discussed the NML probability mass function p_k^{NML} in Section 3.1, and the densities for different model class types f_{t_i} in Section 3.2.

4.1 Gaussian Clusters and Noise

Finding clusters that are modelled with normal distributions is one of the most fundamental clustering problems. We concentrate here and in the following sections on hard clustering, in which every point is assigned to exactly one cluster in contrast to the probabilistic assignments of a soft clustering. Choosing an appropriate number of clusters is a typical example of a model class selection problem. A practical issue is that models with Gaussian clusters do not often fit real world data very well, and a number of attempts has been made to augment the usability of such models, in particular to make them more robust for noisy data (see e.g. [8]).

In Paper II we increase the robustness of the model with a classical method: by adding a component with a uniform distribution. However, the criterion that is used for model selection is novel, and it uses the type of NML based densities that were discussed in Section 3.2. The normal distributions in the model classes were either axis-aligned (with independent coordinates) or spherical. We do not consider the more complicated setting with general normal distributions, but we refer the interested reader to [12, 13] for NML calculations for restricted data in that case. The multinomial NML for a clustering sequence from Section 3.1 would be a correct choice in this context, but Paper II, as well as Paper III, uses the older, general multinomial NML.

To demonstrate the performance of our MDL criterion, we also made experiments with synthetic data using our search heuristic, which is described below. Even if the heuristic is simple, it was effective in finding the intuitively correct clustering when the data were generated according to a mixture distribution with several normal components and one uniform component. During the search procedure the number of clusters diminished from a predetermined number to one, and the model class selection is done during that process. This is a major difference for example to the famous expectation maximization (EM) algorithm [9] which operates within one model class.

The basic idea of the heuristic is that when the data consist of k dense clusters and relatively uniform background noise, running the EM algorithm with more than k components often results in a clustering in which the original clusters are captured quite well but the noise component is covered with several clusters. Then it is possible to prune the inessential clusters and assign their points to a single uniform noise cluster. The outline of the search method is the following:

1. Find an initial clustering with m clusters.

2. If the model does not have a uniform component, add an empty uniform cluster.

3. Sort the points according to ascending density in the current model. Let the sorted sequence be $(y_1, y_2, \ldots, y_n)$.

4. For all $i \in \{1, 2, \ldots, n\}$:

   (a) Move y_i to the uniform cluster.
   (b) Update the parameters of the clusters that were changed; if a cluster became empty, decrement the number of clusters in the model by one. Do not change the cluster assignments of other points than y_i.
   (c) Calculate and store the MDL of the new clustering.

5. Return the clustering that had the smallest MDL.

The initial clustering can be found with any clustering method. We used a hard clustering variant of the EM algorithm, experimenting with different numbers of normal components and either one or zero uniform components. It is commonly known that the EM algorithm is sensitive to the choice of initial clusters at the very beginning of the algorithm. For that purpose we used a random seeding algorithm from [3]. But because that seeding method is designed for the k-means clustering algorithm, there is probably place for improvement. Because of the randomness of the seeding, we repeated the whole clustering procedure 20 times for each data set and parameter combination and picked finally the clustering with the smallest MDL.

The number of the components m in the initial clustering affects the outcome, and there are many possible strategies for trying out different values. In our experiments with data that were generated from a source corresponding one of the model classes, the quality of the final clustering was not sensitive to the choice of m in the beginning, as long as m was suitably large compared to the number of components in the generating model. This is an advantage compared to the regular EM algorithm, as less repeats with different values of m are needed for finding a good clustering.

There are several ways how the basic search algorithm could be further varied. Firstly, for reasons of computational efficiency, we used a fixed order in which the points were moved to the uniform cluster. An interesting variation would be to update the order according to the changed model. Secondly, the finite support of the uniform cluster can cause difficulties that could presumably be avoided by using some other flat distribution. As described above, if the initial clustering did not have a uniform component, we did not force in the beginning of step 4 that the points determining the smallest enclosing box of the data should belong to the uniform cluster. The addition of a single point to the uniform cluster at some point during step 4 can potentially increase the code length for the points of the uniform cluster by a large amount. Because the uniform cluster never shrinks, the method is sensitive to the initial clustering that dictates the order in which the points are handled in step 4.

4.2 Clustgrams: an Extension to Histograms

Density estimation [42, 14] is a fundamental problem in statistical inference and machine learning. One of simplest and most widely-used methods is a histogram. Given a sample of one-dimensional observations $\mathbf{x} \in \mathbb{R}^n$, the goal of *histogram density estimation* is to learn a piecewise constant density that fits the data best according some criterion. We seldom assume that the data are actually produced by a data generating source with an underlying histogram density. Especially when the sample size is small, there are thus some obvious problems with the histogram model. For example, the support of a histogram density is finite, and there is always a discontinuity between two adjacent histogram bins, which can make the estimated density rough.

A natural way to extend histograms is to allow more bin types beside the uniform one. Our *clustgram* framework from Paper III allows arbitrary densities as components, if we are able to calculate the code lengths corresponding to data in the clusters. In Paper III our density selection includes seven one-dimensional densities types from the location-scale family (see Section 3.2). As a model class selection criterion we use the length of the code word that describes the number of clusters, the assignments of points to clusters, the types of the clusters and the data. The form of the corresponding density is given in Equation (4.1). As we operate in one-dimensional space, we can use an efficient search algorithm. If the number of clusters is in the set $\{1, 2, \ldots, K\}$, and the length of the data sample is n, we can find the optimal non-overlapping clustering in $O(Kn^2)$ time with a simple dynamic programming algorithm (using the principle described in [15, Section 4.1]). By non-overlapping we mean the following: it holds for all pairs of clusters (i, j), where $i \neq j$, that either all the points in i are smaller than the points in j, or vice versa. It should be noted, however, that our criterion for model selection in itself is applicable to overlapping cluster assignments too.

4.3 Density and Entropy Estimation Using Histograms

We compare empirically different penalized maximum likelihood methods for density and entropy estimation in Paper V. Here, we give first a slightly more detailed description of the new NML density than in the original paper. Then in Subsection 4.3.2 we discuss the results of the empirical tests.

4.3.1 NML Histogram With k Non-Empty Bins

Kontkanen and Myllymäki have introduced an NML histogram [18] in which the number and locations of cut points between histogram bins are optimized. Each model class is associated with k cut points between a fixed minimum and maximum. Depending on the data, some of the $k+1$ bins in the resulting histogram may be empty. We design in Paper V a more clustering-oriented NML histogram selection criterion. There we optimize the borders of non-empty bins instead of cut points between the bins. Thus a histogram model class with k non-empty bins can be considered as a constrained collection of mixture densities that have k uniform components with non-zero weights. We also included the grid determining the set of potential bin borders to the optimized elements.

Let $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ be a data sequence. Assume that $\min(\mathbf{x}) < \max(\mathbf{x})$, and let $a = \min(\mathbf{x})$ and $c = \max(\mathbf{x})$. For all $m \in \{1, 2, \dots\}$, let a regular grid with m intervals be $G_m = \{a, a+w, a+2w, \dots, c-w, c\}$ where $w = (c-a)/m$. Let $\mathcal{G} = \{G_i \mid i \in J\}$ be the set of possible grids, where $J \subset \{1, 2, \dots\}$. Let us now fix the number of grid intervals m and study histograms defined on the grid G_m. We can define the choice of k non-empty bins on G_m by giving the bin borders $(b_{2j-1}, b_{2j}) \in G_m \times G_m$ for every bin index $j \in \{1, 2, \dots, k\}$. Note that here $b_1 = a$ and $b_{2k} = c$. We assume that $b_{2j-1} < b_{2j}$ for all $j \in \{1, 2, \dots, k\}$ and $b_{2j} \leq b_{2j+1}$ for all $j \in \{1, 2, \dots, k-1\}$. Using the length w as unit, the width of bin j is $w_j = (b_{2j} - b_{2j-1})/w$, and the width of the interval between bins j and $j+1$ is $e_j = (b_{2j+1} - b_{2j})/w$. As we know that $\sum_{i=1}^{k-1}(w_i + e_i) + w_k = m$, or equivalently $\sum_{i=1}^{k-1}(w_i + (e_i + 1)) + w_k = m + k - 1$, the number of different choices of k non-empty bins on G_m is the same as the number of different compositions of the number $m + k - 1$ into $2k - 1$ positive terms, which is given by the binomial coefficient $\binom{m+k-2}{2k-2}$.

We give next the joint density of the grid $G_m \in \mathcal{G}$, the number of non-empty bins $k \in \{1, 2, \dots, \min\{m, n\}\}$, the bin borders $\mathbf{b} = (a, b_2, b_3, \dots, b_{2k-1}, c) \in (G_m)^{2k}$, and the data $\mathbf{x}$. Assume that $a < b_2 \leq b_3 < b_4 \leq \cdots \leq b_{2k-1} < c$. Let the intervals corresponding to the non-empty bins be $I_1 = [a, b_2[, I_2 = [b_3, b_4[, \dots, I_k = [b_{2k-1}, c]$. For all $j \in \{1, 2, \dots, k\}$, let $v_j = b_{2j} - b_{2j-1}$ be the bin width and let $n_j = \sum_{i=1}^{n} \mathbf{1}_{I_j}(x_i)$ be the number of points falling into bin j. Finally, let $\mathbf{c} = (c_1, c_2, \dots, c_n) \in \{1, 2, \dots, k\}^n$ denote the sequence of bin assignments. Notice that because of the bin encoding scheme, $\mathbf{c}$ contains now more information than the clustering sequences we discussed in Section 3.1. Especially, it can now be that $c_1 \neq 1$.

Using the notation from Section 3.1, the NML density of the sequence $\mathbf{x}$

given the histogram $\mathbf{b}$ with k non-empty bins is

$$\begin{aligned} f_{\mathrm{NML}}(\mathbf{x};\, \mathbf{b}) &= \frac{P_k(\mathbf{x};\, \hat{\mathbf{p}}_k(\mathbf{x}))}{\sum_{\mathbf{y}\in D'_k} P_k(\mathbf{y};\, \hat{\mathbf{p}}_k(\mathbf{y}))} \prod_{i=1}^{k} \left(\frac{1}{v_i}\right)^{n_i} \\ &= \frac{1}{\mathcal{C}_1(k,n)} \prod_{i=1}^{k} \left(\frac{n_i}{n}\frac{1}{v_i}\right)^{n_i} \end{aligned}$$

if $\sum_{i=1}^{k} n_i = n$, and $f_{\mathrm{NML}}(\mathbf{x};\, \mathbf{b}) = 0$ otherwise. Probability mass function P_k was defined in (3.2) and normalizing constant $\mathcal{C}_1(k,n)$ in (3.4), and $D'_k = \{\mathbf{y} \in \{1,2,\ldots,k\}^n \mid n_1(\mathbf{y}),\ldots,n_k(\mathbf{y}) \geq 1\}$. A significant difference to the clustering applications in Sections 4.1 and 4.2 is that the clusters in a histogram cannot overlap.

We model the parameter combination $(G_m, k, \mathbf{b})$ using uniform distributions over the corresponding sets. Thus, we define the density of $(G_m, k, \mathbf{b}, \mathbf{x})$ as

$$f(G_m, k, \mathbf{b}, \mathbf{x}) = \frac{1}{|\mathcal{G}|}\frac{1}{\min\{m,n\}}\binom{m+k-2}{2k-2}^{-1} f_{\mathrm{NML}}(\mathbf{x};\, \mathbf{b})\,. \tag{4.2}$$

Finding the best histogram requires finding the optimal number of non-empty bins for each grid $G_m \in \mathcal{G}$ and choosing then the one maximizing the density in (4.2).

Optimizing the places of non-empty bins instead of cut points also has a slight benefit for a practical implementation. In order to achieve $O(Kn^2)$ complexity in the dynamic programming algorithm (see Subsection 4.2), we must build a table containing information of all possible single bins that can exist in the optimal solution. The grid determines where the bin borders can be put, setting also a minimum width for a single bin. When the number of data points is n, there are at most $n+(n-1)+\cdots+1 = (n^2+n)/2$ possible non-empty bins that have to be considered for an optimal histogram. The limit is achieved when all the data points belong to different grid intervals. However, when the places of cut points are optimized as in [18], a table built in a straightforward way can contain as many as $(2n-1)+(2n-2)+\cdots+1 = 2n^2 - n$ entries.

We have used in both cases the fact that the bins of an optimal histogram are in a certain sense as compact as possible. Let us call a grid interval empty if no data point belongs to that interval. In a histogram with optimal non-empty bins, it is clear that in bin $[c,d[$ the first and last grid intervals are non-empty. In other words, intervals $[c,\ c+v[$ and $[d-v,\ d[$ are non-empty, where v is the length of a grid interval.

When cut points are optimized, the cut points are as close as possible to the data points in the following sense: If cut point y lies between two empty grid intervals in an optimal histogram, then y lies between two empty histogram bins as well. There is a simple proof for this in Appendix A.2.

4.3.2 Empirical Comparison of Four Methods

In Paper V we compare empirically the performance of four histogram methods and the clustgram (Subsection 4.2) in density and entropy estimation. The histogram methods are

- Method NML-1, the NML histogram by Kontkanen and Myllymäki [18],
- Method NML-2, the new NML histogram variant (Subsection 4.3.1),
- Method RMG by Rozenholc, Mildenberger and Gather [40],
- Method MRT by Menez, Rendas and Thierry [25].

When there is no prior information about the range of the data, many authors suggest building the histogram between the minimum and maximum of the data [18, 5, 40]. We follow that simplistic convention, even if the approach is slightly problematic in information-theoretic sense. An assumption that the receiver would know the minimum and maximum values of the data with great precision before seeing the message is usually unrealistic. Of course, the range of the data could be transmitted as the first part of a traditional two-part message. Then, it would be possible to optimize the precision of the parameters, because using more bits for parameters allows more effective encoding of the rest of the message. Our intuition is that the performance improvements which could be gained by this optimization would in practice often not be significant enough to compensate the additional complexity of the method. However, in [37] it is assumed that $\min(\mathbf{x}) = 0$, and the maximum of the histogram range is explicitly encoded.

The methods can be divided into two groups based on the grid they use. Methods NML-1 and NML-2 use a regular grid between the minimum and maximum values, methods RMG and MRT use the data points as a grid. In the latter approach, the data points $(x_1, x_2, \ldots, x_n)$ determine a set $\mathcal{A} = \{[y_1, y_2],]y_2, y_3], \ldots,]y_{m-2}, y_{m-1}],]y_{m-1}, y_m]\}$ where $y_1, y_2, \ldots, y_m$ are the elements of set $\{x_1, x_2, \ldots, x_n\}$ in ascending order. The histogram bins are then formed by combining adjacent intervals from $\mathcal{A}$. A direct consequence is that every bin produced by methods RMG and MRT includes at least one data point.

Another categorization criterion for the methods is whether cut points or non-empty clusters are optimized. Note for example that when the minimum and maximum points are know, a histogram with four bins has three cut points and from two to four non-empty clusters. NML-1, RMG and MRT optimize the places of cut points, whereas NML-2 and the clustgram optimize the borders of non-overlapping clusters (the search algorithm does not find overlapping clusters even if the cost function of the clustgram would allow them).

NML-1 requires choosing the grid as a parameter, and the clustgram method has an accuracy parameter ϵ as well as parameters a and b for the prior density of the location. Although these two methods do not seem to be very sensitive to the choice of parameters, and reasonable values can thus be found relatively easily in practice, their dependency on parameters can still be seen as a weakness compared to the parameterless methods RMG and MRT. Method NML-2, which chooses the best grid from a given set, lies in the middle of parameter-dependent and parameter-free methods. Modifying NML-1 to optimize the grid similarly to NML-2 would be trivial (see (4.2)). Still, we implemented NML-1 using the original code length from [18]. For more details about choosing the parameters for the testing procedure, we refer to Paper V.

For evaluation of the methods, synthetic data were drawn randomly from 50 different distributions, which are listed in Appendix A.3 and illustrated in Figures A.3–A.12. They include the 15 normal mixtures from [24], 2 normal mixtures from [18], mixtures with normal, exponential, uniform and triangular components, as well as different unimodal distributions. The accuracy of density estimation was measured by integrating numerically the squared Hellinger distance $h^2(f,g) = \int (\sqrt{f(x)} - \sqrt{g(x)}\,)^2 \,\mathrm{d}x$ between the known source distribution f and the found model g. The entropy of the source and the model were computed analytically if possible, otherwise by numerical integration.

Method MRT favoured solutions with relatively few bins, which often led to slightly worse estimates compared to other methods. The performance of clustgram method was twofold: if the mixtures consisted of clearly separated components that matched the component types of the clustgram, clustgram often clearly outperformed other methods in density estimation. However, the estimation accuracy of the clustgram was with some strongly overlapping distributions only moderate. Method RMG was a solid performer that tended to choose solutions with more bins than the other histogram methods.

All methods behaved soundly in the sense that the estimation accuracy improved as the sample size increased. There were only two notable exceptions to that rule. One was clustgram method with a symmetric triangular distribution as the data source (density no. 9 in Appendix A.3, Example 3 in Paper V). The second case, which was apparently caused by the implementation, was NML-1 and NML-2 with a heavy-tailed Pareto distribution (density no. 6, see explanation in Appendix A.3).

Interestingly, when we let the set of possible grids for NML-2 to include only one grid with interval length 0.02, method NML-2 was almost always worse in density estimation than NML-1 with the same grid. But using the grid optimization procedure described in Paper V, the results improved significantly. The estimates of NML-2 were especially good with many ragged multimodal distributions that seemed to cause difficulties for the other methods. Interesting examples in particular from the entropy estimation point of view include densities 21, 41, 47, and 48.

Chapter 5

Conclusion

The contributions of this thesis are related to normalized maximum likelihood distributions and their applications. One of the most difficult problems with the NML is the infinite parametric complexity of many relevant model classes. The most obvious solutions, limiting the range of the parameters or the data, induces practical problems. What should we for example do if we unexpectedly encounter a sample that we do not have a code word for? If we do not want to discard the sample completely, we have to change the code book. That leads to an idea of having a library of code books and a separate code for the code book names. However, that system has some obvious problems. One weakness of the design is the concept of discrete code books which usually leads to a discontinuous code length function.

In order to provide a better alternative, we constructed in Papers I, II and III distributions enabling the encoding of all possible data in a sensible way with several model classes: normal, half-normal, uniform, exponential, and Laplace. The distributions of these model classes have simple closed-form maximum likelihood parameter estimators, which is essential for the derivation of the results. If nothing is known about the data and the model classes are not restricted in any way, the NML distributions do not exist in these cases. Although there are thus no regret-minimizing distributions, we still considered the regret as the main criterion for our code design. Our distributions are based on the assumption that we have some very general preconceptions about how the data will be. The preconceptions are expressed with three parameters, and the regret grows very slowly when we encounter data that are more and more surprising. All data are encodable with our density, which makes it more practical than the pure NML.

In addition to the contributions related to the infinite parametric complexity problem, we showed in Paper IV how to calculate efficiently the NML distribution for the multinomial model class in a configuration that is natural for clustering applications (it turned out later that the main result of Paper IV was published previously in [26]). In most applications, we do not expect that using this particular model class definition instead of the general multinomial model class would lead to noticeably different results. However, the calculation of the corresponding NML distribution is equally demanding in both cases, so there are no practical disadvantages for using the newer code length function.

Another contribution of theoretical nature, which is not directly linked to other contents of the thesis, is in explained in Appendix A.1. The unsmooth diminishing of the well-known prior for integers may be considered as a small flaw for applications. Therefore, we propose a variant of the original prior with more regular behaviour.

The qualities of the new code lengths were demonstrated with clustering-based applications, for which Section 4 was dedicated. In Paper II we discussed a group a clustering methods that took advantage of the NML based code lengths. A common feature for the methods was that the number of clusters and thus the complexity of the model changed during the clustering process. In tests with synthetic data, the new clustering heuristics were successful in finding the original cluster structure.

The clustgram from Paper III is an extension to the histogram and a straightforward application of the new code lengths. Its performance in density and entropy estimation was evaluated in Paper V against two NML histograms, one of which was a novel contribution, and two other penalized histogram methods. The variety of different component types in the clustgram was a major benefit for density estimation in some cases. On the other hand, the clustgram was generally not among the best methods in entropy estimation. The new computationally demanding NML histogram (called "NML-2" in the paper) performed especially well with complex ragged mixture densities.

Appendix

A.1 Code Lengths for Natural Numbers

How to encode integers in the set $\mathbb{N}_+ = \{1, 2, \dots\}$ is a fundamental coding problem. It can arise for example in model selection problems when the number of model classes is infinite: we may want to encode the index of the model class explicitly without favouring any class too strongly. If the index set were finite, our choice would be most likely the uniform distribution. But what distribution over a countably infinite set could be a conceptual counterpart of the uniform distribution?

We propose a modified version of Rissanen's prior for encoding of positive integers. In the original distribution, the probabilities of the first few integers diminish at an uneven rate, which we want to avoid. Our modified prior also has a parameter with which the distribution of probability mass can be tuned. Notice that there are also other codes that share the same kind of assymptotic properties as Rissanen's prior: the codes for decision trees in [28] and [48] can be used for coding of integers as well.

In the following, we let log denote logarithm to base 2. Rissanen proposes in [30] the following probability for encoding of integers: $P_0 : \mathbb{N}_+ \to]0, 1[$, $P_0(n) = (c_o n \cdot h(n))^{-1}$ where $c_0 \approx 2.865064$ is a normalizing constant and

$$h(x) = \begin{cases} 1 & \text{if } \log x \leq 1 \\ (\log x)\; h(\log x) & \text{if } \log x > 1\,. \end{cases}$$

The choice of P_0 is motivated by the fact that it is hard to find practical probability mass functions that would diminish asymptotically as slowly as P_0. Rissanen calls P_0 *a universal prior for the integers*, and Grünwald adopts the terminology in [11]. One should note that the probabilities which P_0 assigns to different integers are not "universal" themselves. As we shall soon see, there are an infinite number of probability mass functions that could be called universal priors for integers as well as P_0.

There are some regularity properties that would be advantageous for an integer prior P. First, we expect P to be monotonically decreasing in $\mathbb{N}_+$. Then, at least, it would be desirable if the mapping $n \mapsto -\log P(n+1) - (-\log P(n))$ were monotonically decreasing as well (or $n \mapsto P(n+1)/P(n)$ were monotonically increasing). This kind of smoothness ensures that P does not cause artificial thresholds in model class selection.

Rissanen's prior P_0 clearly fails to fulfil the second condition. Let the right-associative operator $x \uparrow y$ denote x^y. Let $x \uparrow\uparrow 0 = 1$, and let $x \uparrow\uparrow y = \underbrace{x \uparrow x \uparrow \ldots \uparrow x}_{y \text{ copies of } x}$ for $x > 0$, $y \in \mathbb{N}_+$. If we write $L_0(n) = -\log P_0(n)$, it is easy to verify that the condition $L_0(n+1) - L_0(n) < L_0(n) - L_0(n-1)$ is false when $n \in \{2, 4, 16, 65536\} = \{2 \uparrow\uparrow 1,\ 2 \uparrow\uparrow 2,\ 2 \uparrow\uparrow 3,\ 2 \uparrow\uparrow 4\}$. The increase of the code lengths for the first few integers is therefore quite uneven (see Figures A.1 and A.2). The problem could be circumvented for example by shifting the origin and renormalizing accordingly.

Instead of directly manipulating P_0, we derive a new probability mass function for the natural numbers by integrating a related density for the non-negative real numbers. The resulting function is smoother than P_0 and it has one parameter. In Paper II we made a parameterized version of the prior for reals from [30]. We use here our prior for reals but with a slightly simpler parameterization, which we explain next.

For $x \in [0, \infty[$, we define the density

$$f_{\mathbb{R}_+}(x;\, k) = \frac{1 - \ln 2}{(\ln 2)^k} \frac{1}{(x+b)\, h(x+b)}$$

where $k \in \mathbb{N}_+$ and $b = 2 \uparrow\uparrow (k-1)$. Note that the derivative of $f_{\mathbb{R}_+}(\cdot;\, k)$ is discontinuous at $(2 \uparrow\uparrow m) - b$, $m \in \{k, k+1, \ldots\}$. Now we can derive our prior for the natural numbers by integrating $f_{\mathbb{R}_+}$. We write a multiple logarithm as

$$\log^{(k)} x := \underbrace{\log \log \ldots \log}_{k \text{ copies}} x$$

and we let

$$\log^{\diamond} x := \max \Big\{ \{0\} \cup \{k \in \mathbb{N}_+ \mid \log^{(k)} x > 1\} \Big\}$$

denote a logarithmic order of magnitude of x. Let $n \in \{b+1, b+2, \ldots\}$ and let $m = \log^{\diamond} n$. It follows that $(2 \uparrow\uparrow m) + 1 \leq n \leq 2 \uparrow\uparrow (m+1)$. If $n - 1 < x < n$, then $2 \uparrow\uparrow m < x < 2 \uparrow\uparrow (m+1)$ and

$$D_x \Big((\ln 2)^{m+1} \log^{(m+1)} x \Big) = \frac{1}{x\, h(x)}\,.$$

Thus, we get the integral

$$\int_{n-1}^{n} \frac{1}{x\,h(x)}\,\mathrm{d}x = (\ln 2)^{m+1}\Big(\log^{(m+1)} n - \log^{(m+1)}(n-1)\Big)\,.$$

This yields the desired prior for all $n \in \mathbb{N}_+$,

$$\begin{aligned} P_{\mathbb{N}_+}(n;\,k) &= \int_{n-1}^{n} f_{\mathbb{R}_+}(x;\,k)\,\mathrm{d}x \\ &= \int_{n-1}^{n} \frac{1-\ln 2}{(\ln 2)^k}\,\frac{1}{(x+b)\,h(x+b)}\mathrm{d}x \\ &= \frac{1-\ln 2}{(\ln 2)^k}\int_{n+b-1}^{n+b}\frac{1}{y\,h(y)}\,\mathrm{d}y \\ &= (1-\ln 2)(\ln 2)^{j+1-k}\Big(\log^{(j+1)}(n+b) - \log^{(j+1)}(n+b-1)\Big) \end{aligned}$$

where $b = 2 \uparrow\uparrow (k-1)$ and $j = \log^{\diamond}(n+b)$.

Figure A.2 illustrates that $P_{\mathbb{N}_+}(\cdot;\,k)$ decreases more smoothly than P_0 with the first few integers even if we do not achieve the condition that $n \mapsto P_{\mathbb{N}_+}(n+1;\,k)/P_{\mathbb{N}_+}(n;\,k)$ were monotonically increasing everywhere. Parameter k controls the heaviness of the tails of $P_{\mathbb{N}_+}$ (see Figure A.1). This addition seems to be necessary for many applications when we think of the fact that $P_0(1) + P_0(2) > 1/2$.

A.2 Optimal Cut Point Placement Between Two Histogram Bins

We prove here a simple result that is essential for an efficient implementation of Kontkanen's and Myllymäki's NML histogram [18]. The grid on which the histogram is built can be fine and contain more intervals than there are data points. However, it is not necessary to consider all possible intervals of the grid as potential bins. We show that if cut point y lies between two empty grid intervals in an optimal solution with a fixed number bins, then y is a cut point between two empty histogram bins. By merging these two adjacent empty bins, we have a solution with less bins and a shorter code length.

Let y be located between two empty grid intervals in an optimal histogram. Assume first that exactly one of the bins adjacent to y is empty. Now we can replace cut point y with cut point y' so that the non-empty bin becomes

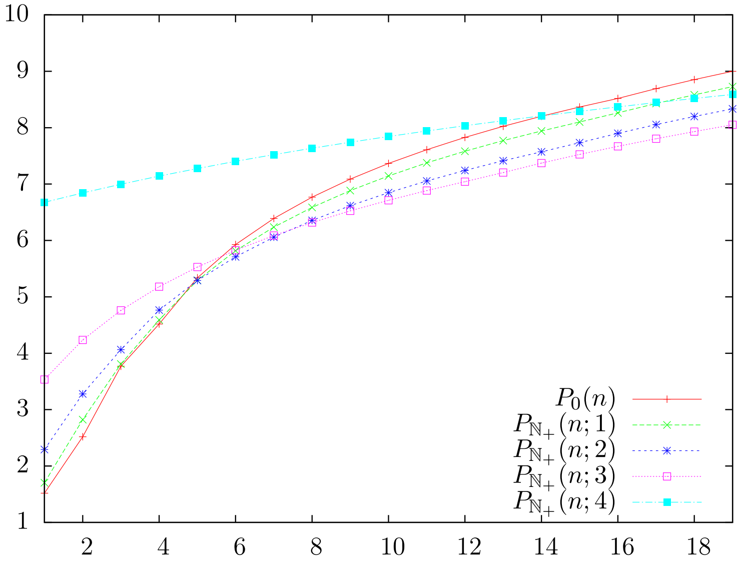


Figure A.1: Code lengths for integers 1, 2, . . . , 19 according to Rissanen's prior for integers P_0 and the new integer prior $P_{\mathbb{N}_+}$.

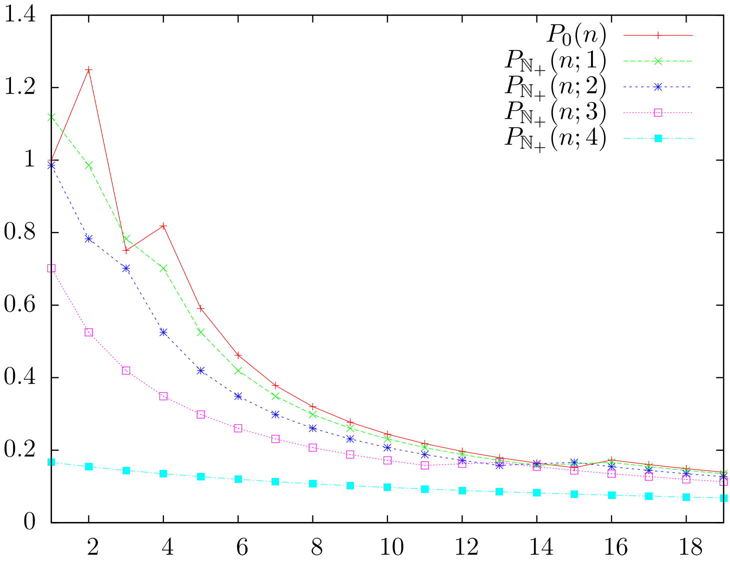


Figure A.2: Code length differences $L(n+1) - L(n)$ for $n \in \{1, 2, \ldots, 19\}$. Comparison between Rissanen's integer prior P_0 and the new prior $P_{\mathbb{N}_+}$.

narrower. That clearly makes the likelihood of the histogram larger, which leads to a contradiction.

Let us next consider the placement of a cut point between two non-empty bins. Assume that there are r points in the interval $[a, b[$, 0 points in the interval $[b, c[$, and s points in the interval $[c, d[$, where $r, s, \geq 1$ and $a < b < c < d$. We want to find the optimal cut point $y \in [b, c]$ so that the intervals $[a, y[$ and $[y, d[$ form two histogram bins. Let n be the total number of points (where possibly $r + s \neq n$). The contribution of the two bins to the likelihood of the histogram is as a function of the cut point

$$\begin{aligned} g(y) &= \left(\frac{r}{n} \cdot \frac{1}{y-a}\right)^r \left(\frac{s}{n} \cdot \frac{1}{d-y}\right)^s \\ &= \frac{r^r s^s}{n^{r+s}} \cdot \frac{1}{(y-a)^r} \, \frac{1}{(d-y)^s} \, . \end{aligned}$$

The derivative of g is

$$g'(y) = \frac{r^r s^s}{n^{r+s}} \cdot \frac{rd + sa - (r+s)y}{(y-a)^{r+1}\,(d-y)^s\,(y-d)} \, .$$

It is easy to see that the denominator of $g'(y)$ is negative and that $rd + sa - (r + s)y$ decreases as y grows. So the sign of $g'(y)$ can change only once between b and c, and in that case $g'(b) < 0$ and $g'(c) > 0$. Therefore, the maximum value of g in the interval $[b, c]$ is either $g(b)$ or $g(c)$. By choosing b and c in a suitable way, it follows easily that an optimal cut point between two non-empty bins cannot lie between two empty grid intervals.

A.3 Histogram Methods: Results to Section 4.3.2

Random samples for the evaluation of the methods in Section 4.3.2 were generated from 50 distributions listed below. We use the following notations:

- normal distribution with mean μ and variance σ^2:

$$\varphi(\mu, \sigma^2)(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

- shifted exponential distribution with origin α and mean β:

$$\mathrm{Exp}_{+}(\alpha, \beta)(x) = \frac{e^{-(x-\alpha)/\beta}}{\beta}$$

if $x \geq \alpha$, otherwise $\mathrm{Exp}_{+}(\alpha, \beta)(x) = 0$

- shifted mirrored exponential distribution with origin α and mean β:

  $$\mathrm{Exp}_{-}(\alpha,\beta)(x) = \frac{e^{-(\alpha-x)/\beta}}{\beta}$$

  if $x \le \alpha$, otherwise $\mathrm{Exp}_{-}(\alpha,\beta)(x) = 0$.

- uniform distribution in the interval $[a, b]$, where $a < b$:

  $$U([a,b])(x) = \frac{1}{b-a}$$

  if $x \in [a, b]$, otherwise $U([a, b])(x) = 0$.

The densities are listed below. They are also illustrated beside the test results graphs in Figures A.3–A.12. The averages of squared Hellinger distances between the known source density and the estimated densities in 100 test runs are presented in the middle column of Figures A.3–A.12. In the rightmost column, the centre point of an error bar indicates the average estimated entropy in 100 runs, and the total height of a bar corresponds to two sample deviations. The horizontal line indicates the entropy of the source distribution. Nats are used as units for the entropy. The sample sizes on the x axis of the test result graphs are 50, 100, 200, 400, 800, 1600 and 3200.

We comment some striking results here. The estimation accuracy of NML-1 and NML-2 worsens with the Pareto distribution (density no. 6) and the largest sample size probably because of an implementation level detail. With both NML methods, the grid is determined by dividing the range of the data into intervals of fixed length. In the program however, the number of these intervals between the minimum and maximum point was limited to 2 147 483 646, which was the C programming language constant `INT_MAX - 1` in our system. That means, if the range of the data was large enough, the implementation changed the length of the grid intervals. Because of the long tails of the density and the fairly large sample size, the limit was hit in 43% of the test runs with NML-1. NML-2 chose in 10% of the runs a grid in which the number of grid intervals was limited in this way. When these cases were not considered, the estimation accuracy of NML-1 and NML-2 improved with the largest sample size in a similar way as with density no. 5, a Pareto distribution with lighter tails.

List of source distributions

Simple unimodal distributions

1. Standard normal distribution (density no. 1 in [24]): $f = \varphi(0, 1^2)$.
2. Normal distribution with $\mu = 0$, $\sigma^2 = 10^2$: $f = \varphi(0, 10^2)$.
3. Chi-squared distribution with 2 degrees of freedom: $f(x) = 1/2 \cdot \exp(-x/2)$ for $x \geq 0$.
4. Chi-squared distribution with 10 degrees of freedom: $f(x) = 1/768 \cdot x^4 \exp(-x/2)$ for $x \geq 0$.
5. Pareto distribution: $f(x) = x^{-2}$ for $x \geq 1$.
6. Pareto distribution: $f(x) = 1/2 \cdot x^{-3/2}$ for $x \geq 1$.
7. Cauchy distribution with location 0 and scale 1: $f(x) = \pi^{-1}(1+x^2)^{-1}$.
8. Cauchy distribution with location 0 and scale 2: $f(x) = (2\pi)^{-1}(1 + (x/2)^2)^{-1}$.
9. Symmetric triangular distribution, positive in $]-1, 1[$, maximum at 0.
10. Triangular distribution, positive in $]-5, 15[$, maximum at 0.
11. Uniform distribution: $f = U([-1, 1])$.
12. Uniform distribution: $f = U([-20, 20])$.
13. Exponential distribution with mean 1: $f = \mathrm{Exp}_+(0, 1)$.
14. Exponential distribution with mean 10: $f = \mathrm{Exp}_+(0, 10)$.
15. Laplace distribution with mean 0 and scale 1: $f(x) = 1/2 \cdot \exp(-|x|)$.
16. Laplace distribution with mean 0 and scale 8: $f(x) = 1/16 \cdot \exp(-|x|/8)$.

Mixtures of normal and exponential densities

17. Mixture of 2 shifted exponential distributions, one of them mirrored: $f = 0.333 \cdot \mathrm{Exp}_-(-2, 2) + 0.667 \cdot \mathrm{Exp}_+(2, 2)$.
18. Mixture of 2 shifted exponential distributions, one of them mirrored: $f = 0.333 \cdot \mathrm{Exp}_+(-2, 2) + 0.667 \cdot \mathrm{Exp}_-(-2, 2)$.

19. Mixture of 4 shifted exponential distributions: $f = 1/4 \cdot \mathrm{Exp}_+(-5, 1) + 1/4 \cdot \mathrm{Exp}_+(-3, 2) + 1/4 \cdot \mathrm{Exp}_+(0, 1/2) + 1/4 \cdot \mathrm{Exp}_+(5, 4)$.

20. Mixture of 4 shifted exponential distributions, two of them mirrored: $f = 4/10 \cdot \mathrm{Exp}_+(-3, 3) + 2/10 \cdot \mathrm{Exp}_-(-1, 2) + 1/10 \cdot \mathrm{Exp}_+(0, 1/2) + 3/10 \cdot \mathrm{Exp}_-(5, 4)$.

21. Mixture of 8 shifted exponential distributions: $f = 1/12 \cdot \mathrm{Exp}_+(-8, 1) + 1/6 \cdot \mathrm{Exp}_+(-6, 2) + 1/12 \cdot \mathrm{Exp}_+(-4, 1) + 1/6 \cdot \mathrm{Exp}_+(-2, 2) + 1/12 \cdot \mathrm{Exp}_+(0, 1) + 1/6 \cdot \mathrm{Exp}_+(2, 2) + 1/12 \cdot \mathrm{Exp}_+(4, 1) + 1/6 \cdot \mathrm{Exp}_+(6, 2)$.

22. Mixture of 8 shifted exponential distributions, four of them mirrored: $f = 1/12 \cdot \mathrm{Exp}_-(-8, 1) + 1/6 \cdot \mathrm{Exp}_+(-6, 2) + 1/12 \cdot \mathrm{Exp}_-(-2, 1) + 1/6 \cdot \mathrm{Exp}_+(-1, 2) + 1/12 \cdot \mathrm{Exp}_-(0, 1) + 1/6 \cdot \mathrm{Exp}_+(1, 2) + 1/12 \cdot \mathrm{Exp}_-(2, 1) + 1/6 \cdot \mathrm{Exp}_+(3, 2)$.

23. Mixture of a shifted exponential and a normal distribution: $f = 1/5 \cdot \mathrm{Exp}_+(-10, 5) + 4/5 \cdot \varphi(5, 3^2)$.

24. Mixture of a shifted exponential and a normal distribution: $f = 2/5 \cdot \mathrm{Exp}_+(3, 3) + 3/5 \cdot \varphi(10, 3^2)$.

25. Mixture of 2 shifted exponential and 2 normal distributions: $f = 1/10 \cdot \mathrm{Exp}_-(-20, 1) + 3/10 \cdot \varphi(-10, 3^2) + 4/10 \cdot \varphi(0, 2^2) + 2/10 \cdot \mathrm{Exp}_+(8, 5)$.

26. Mixture of 2 shifted exponential and 2 normal distributions: $f = 1/10 \cdot \mathrm{Exp}_-(-8, 1) + 4/10 \cdot \varphi(-6, 3^2) + 3/10 \cdot \varphi(1, 2^2) + 2/10 \cdot \mathrm{Exp}_+(2, 5)$.

27. Mixture of 4 shifted exponential and 4 normal distributions: $f = 0.125 \cdot \mathrm{Exp}_-(-15, 2) + 0.125 \cdot \varphi(-14, 1^2) + 0.125 \cdot \varphi(-10, 2^2) + 0.07 \cdot \mathrm{Exp}_+(-7, 1) + 0.18 \cdot \mathrm{Exp}_-(-3, 4) + 0.15 \cdot \varphi(0, 3^2) + 0.1 \cdot \varphi(5, (1/2)^2) + 0.125 \cdot \mathrm{Exp}_-(10, 2)$.

28. Mixture of 4 shifted exponential and 4 normal distributions: $f = 0.08 \cdot \mathrm{Exp}_-(-14, 2) + 0.17 \cdot \varphi(-14, 1^2) + 0.125 \cdot \varphi(-9, 2^2) + 0.07 \cdot \mathrm{Exp}_+(-9, 1) + 0.18 \cdot \mathrm{Exp}_+(-6, 3) + 0.15 \cdot \varphi(7.5, 2^2) + 0.1 \cdot \varphi(10, 1^2) + 0.125 \cdot \mathrm{Exp}_-(15, 3)$.

Normal mixtures from [24]

29. Mixture of 3 normal distributions (density no. 2 in [24], “skewed unimodal”): $f = 1/5 \cdot \varphi(0, 1^2) + 1/5 \cdot \varphi(1/2, (2/3)^2) + 3/5 \cdot \varphi(13/12, (5/9)^2)$.

30. Mixture of 8 normal distributions (density no. 3 in [24], “strongly skewed”): $f = 1/8 \cdot \sum_{j=0}^{7} \varphi(3 \cdot ((2/3)^j - 1), ((2/3)^j)^2)$.

31. Mixture of 2 normal distributions (density no. 4 in [24], "kurtotic unimodal"): $f = 2/3 \cdot \varphi(0, 1^2) + 1/3 \cdot \varphi(0, (1/10)^2)$.

32. Mixture of 2 normal distributions (density no. 5 in [24], "outlier"): $f = 1/10 \cdot \varphi(0, 1^2) + 9/10 \cdot \varphi(0, (1/10)^2)$.

33. Mixture of 2 normal distributions (density no. 6 in [24], "bimodal"): $f = 1/2 \cdot \varphi(-1, (2/3)^2) + 1/2 \cdot \varphi(1, (2/3)^2)$.

34. Mixture of 2 normal distributions (density no. 7 in [24], "separated bimodal"): $f = 1/2 \cdot \varphi(-3/2, (1/2)^2) + 1/2 \cdot \varphi(3/2, (1/2)^2)$.

35. Mixture of 2 normal distributions (density no. 8 in [24], "skewed bimodal"): $f = 3/4 \cdot \varphi(0, 1^2) + 1/4 \cdot \varphi(3/2, (1/3)^2)$.

36. Mixture of 3 normal distributions (density no. 9 in [24], "trimodal"): $f = 9/20 \cdot \varphi(-6/5, (3/5)^2) + 1/10 \cdot \varphi(0, (1/4)^2) + 9/20 \cdot \varphi(6/5, (3/5)^2)$.

37. Mixture of 6 normal distributions (density no. 10 in [24], density no. 23 in [4], "claw"): $f = 1/2 \cdot \varphi(0, 1^2) + 1/10 \cdot \varphi(-1, (1/10)^2) + 1/10 \cdot \varphi(-1/2, (1/10)^2) + 1/10 \cdot \varphi(0, (1/10)^2) + 1/10 \cdot \varphi(1/2, (1/10)^2) + 1/10 \cdot \varphi(1, (1/10)^2)$.

38. Mixture of 9 normal distributions (density no. 11 in [24], "double claw"): $f = 49/100 \cdot \varphi(-1, (2/3)^2) + 49/100 \cdot \varphi(1, (2/3)^2) + 1/350 \cdot \sum_{j=0}^{6} \varphi((j-3)/2, (1/100)^2)$.

39. Mixture of 6 normal distributions (density no. 12 in [24], "asymmetric claw"): $f = 1/2 \cdot \varphi(0, 1) + \sum_{j=-2}^{2} (2^{1-j}/31)\, \varphi(j + 1/2, (2^{-j}/10)^2)$.

40. Mixture of 8 normal distributions (density no. 13 in [24], "asymmetric double claw"): $f = 46/100 \cdot \sum_{j=0}^{1} \varphi(2j-1, (2/3)^2) + 1/300 \cdot \sum_{j=1}^{3} \varphi(-j/2, (1/100)^2) + 7/300 \cdot \sum_{j=1}^{3} \varphi(j/2, (7/100)^2)$.

41. Mixture of 6 normal distributions (density no. 14 in [24], density no. 24 in [4],"smooth comb"):

$$\begin{aligned} f = \frac{1}{63} \cdot \Big(& 32\, \varphi(-31/21, (32/63)^2) + 16\, \varphi(17/21, (16/63)^2) \\ & + 8\, \varphi(41/21, (8/63)^2) + 4\, \varphi(53/21, (4/63)^2) \\ & + 2\, \varphi(59/21, (2/63)^2) + \varphi(62/21, (1/63)^2) \Big)\,. \end{aligned}$$

42. Mixture of 6 normal distributions (density no. 15 in [24], "discrete comb"):

$$f = 2/7 \cdot \big(\varphi(-15/7,(2/7)^2) + \varphi(-3/7,(2/7)^2) + \varphi(9/7,(2/7)^2)\big) \\ + 1/21 \cdot \big(\varphi(16/7,(1/21)^2) + \varphi(18/7,(1/21)^2) + \varphi(20/7,(1/21)^2)\big)\,.$$

Other mixtures

43. Mixture of 5 normal distributions (from the author of [18]): $f = 0.2 \cdot \varphi(1,1^2) + 0.15 \cdot \varphi(1.75, 0.806^2) + 0.25 \cdot \varphi(3, 1.414^2) + 0.1 \cdot \varphi(5.5, 1.581^2) + 0.3 \cdot \varphi(8, 0.387^2)$.

44. Mixture of 8 normal distributions (from the author of [18]: $f = 0.1 \cdot \varphi(1, 0.592^2) + 0.1 \cdot \varphi(3, 0.949^2) + 0.2 \cdot \varphi(4, 0.671^2) + 0.1 \cdot \varphi(7, 0.922^2) + 0.1 \cdot \varphi(9, 0.742^2) + 0.1 \cdot \varphi(12, 0.949^2) + 0.15 \cdot \varphi(13, 1.095^2) + 0.15 \cdot \varphi(16, 0.949^2)$.

45. Mixture of 2 normal distributions (density no. 21 in [4], "marronite"): $f = 1/3 \cdot \varphi(-20, (1/4)^2) + 2/3 \cdot \varphi(0, 1^2)$.

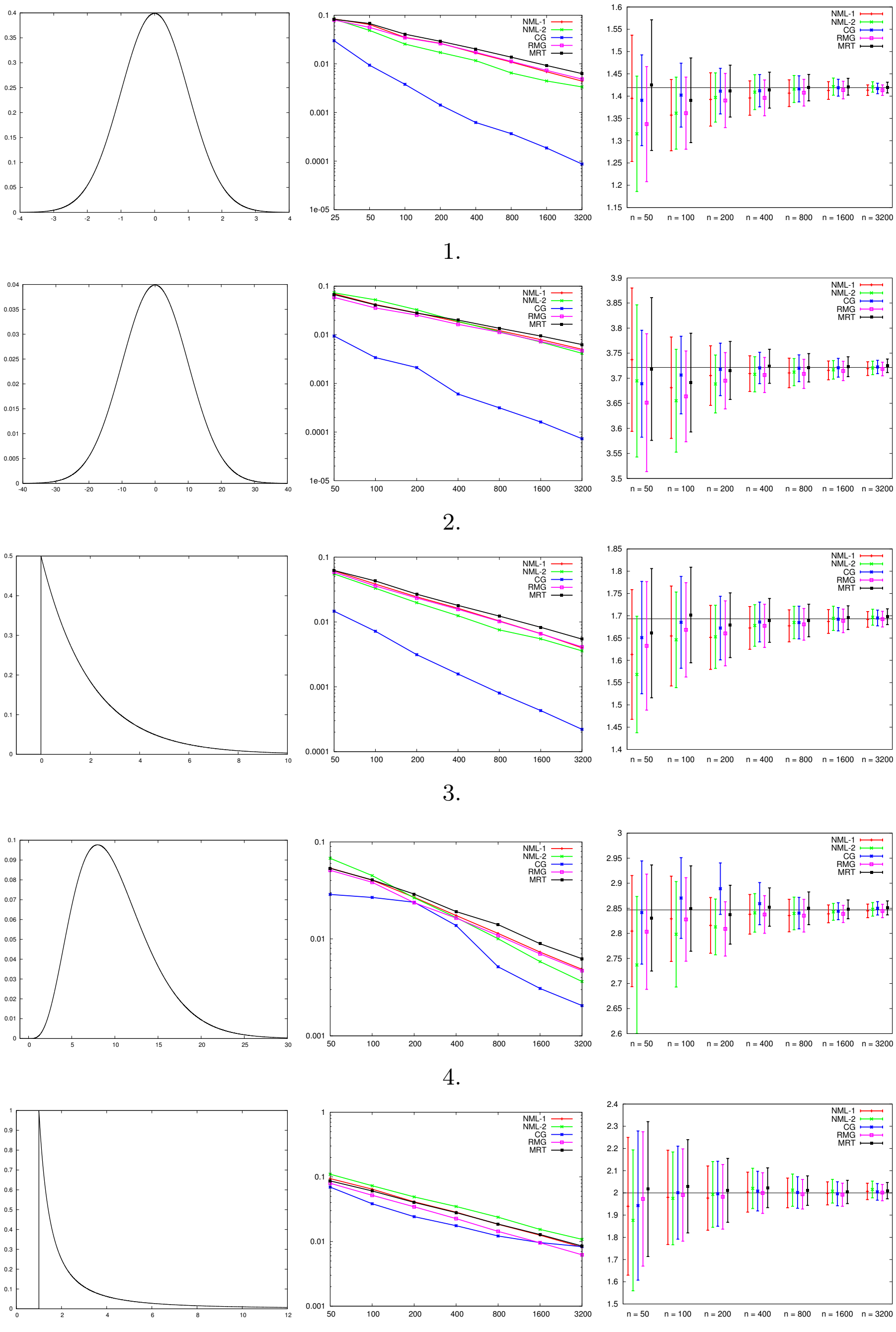
46. Mixture of 3 uniform distributions (density no. 26 in [4], "trimodal uniform"): $f = 1/4 \cdot U([-20.1, -20]) + 1/2 \cdot U([-1, 1]) + 1/4 \cdot U([20, 20.1])$.

47. Mixture 10 triangular distributions (density no. 27 in [4], "sawtooth"): $f = 1/10 \cdot \sum_{j=0}^{9} T(-10 + 2j, -8 + 2j)$ where $T(a, b)$ is a symmetric triangular distribution in interval $[a, b]$.

48. Mixture of 5 uniform distributions: $f = 1/5 \cdot \sum_{j=0}^{4} U([-9 + 4j, -7 + 4j])$.

49. Mixture of 10 normal distributions: $f = 1/10 \cdot \sum_{j=0}^{9} \varphi(9j, 1^2)$.

50. Mixture of 9 triangular distributions: $f = \sum_{j=0}^{2} (1/6 \cdot T(12j, 12j + 2) + 1/18 \cdot T(12j + 4, 12j + 6) + 1/9 \cdot T(12j + 8, 12j + 10))$ where $T(a, b)$ is a symmetric triangular distribution in interval $[a, b]$.



1.

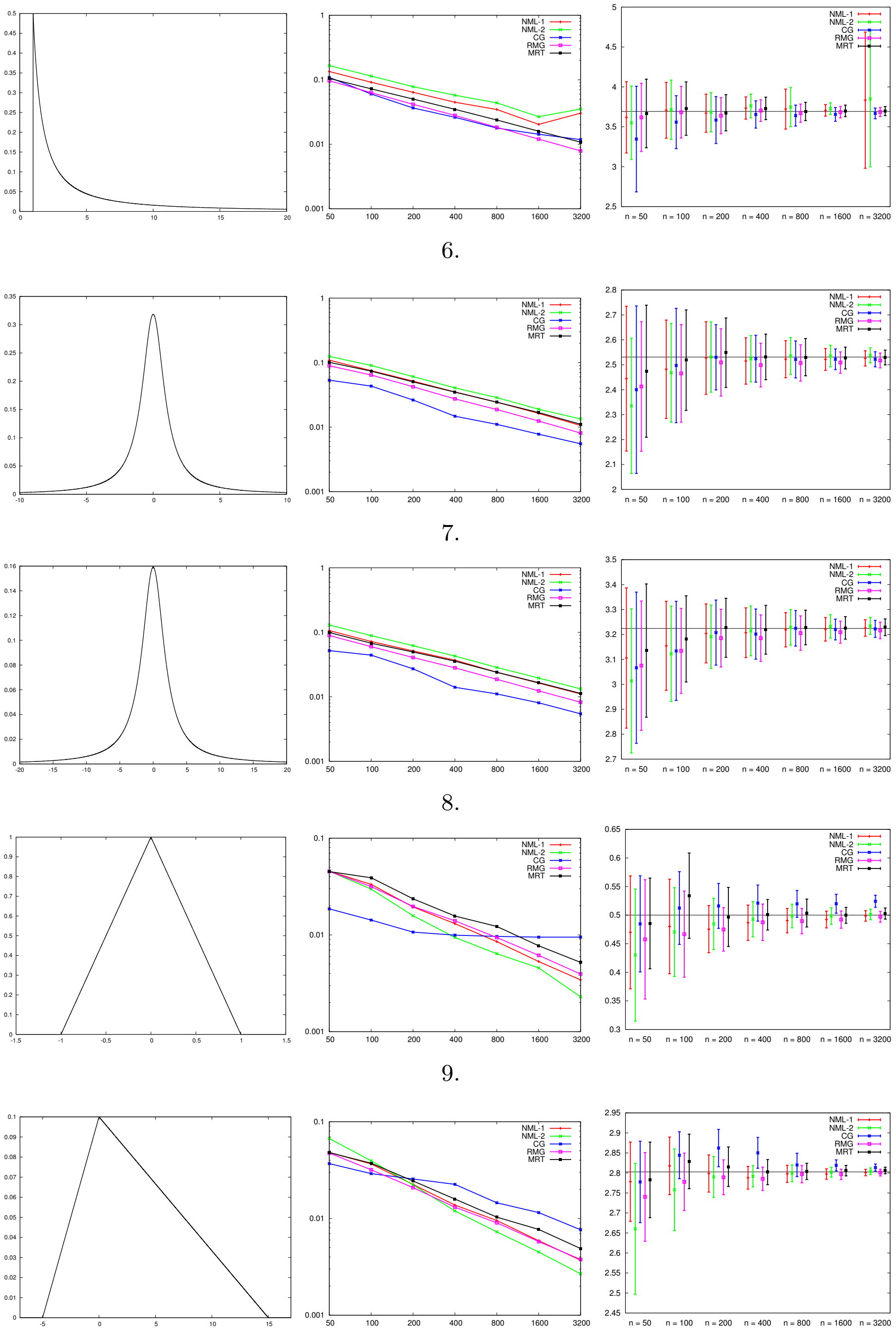
2.

3.

4.

5.

Figure A.3: Density and entropy estimation test results for densities 1–5. The colours for the methods: NML-1, NML-2, CG, RMG, MRT.



6.

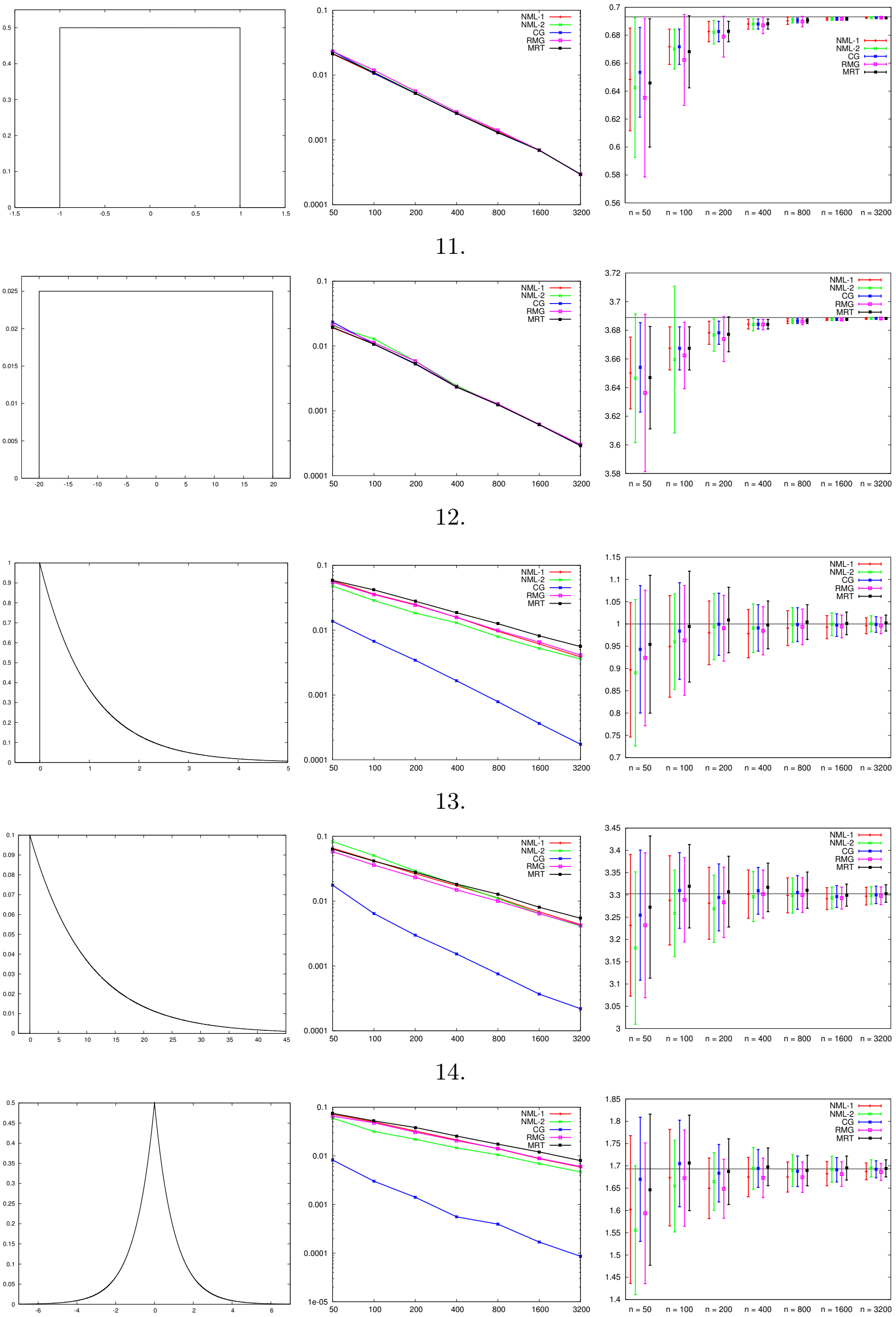
7.

8.

9.

10.

Figure A.4: Density and entropy estimation test results for densities 6–10. The colours for the methods: NML-1, NML-2, CG, RMG, MRT.



11.

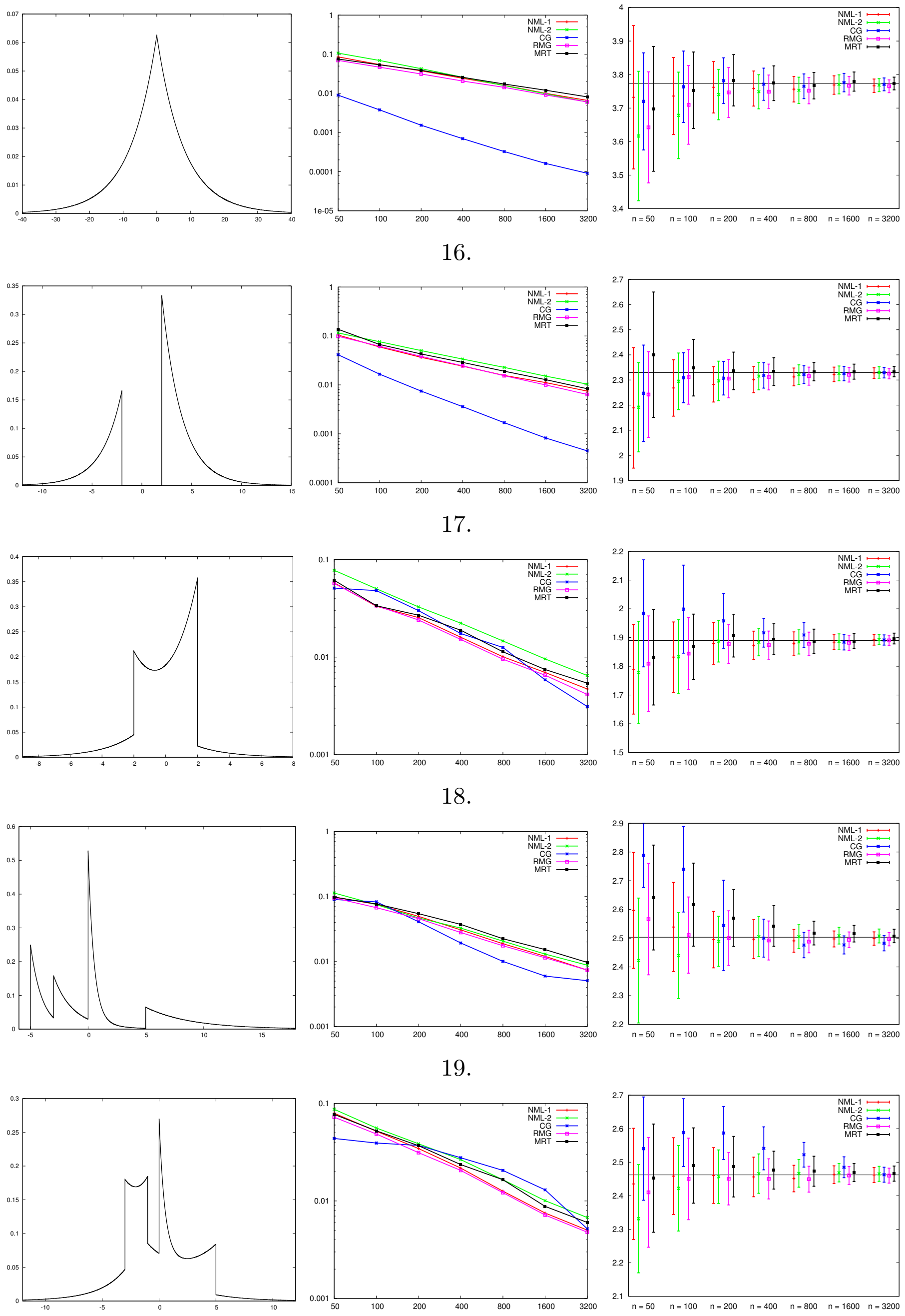
12.

13.

14.

15.

Figure A.5: Density and entropy estimation test results for densities 11–15. The colours for the methods: NML-1, NML-2, CG, RMG, MRT.



16.

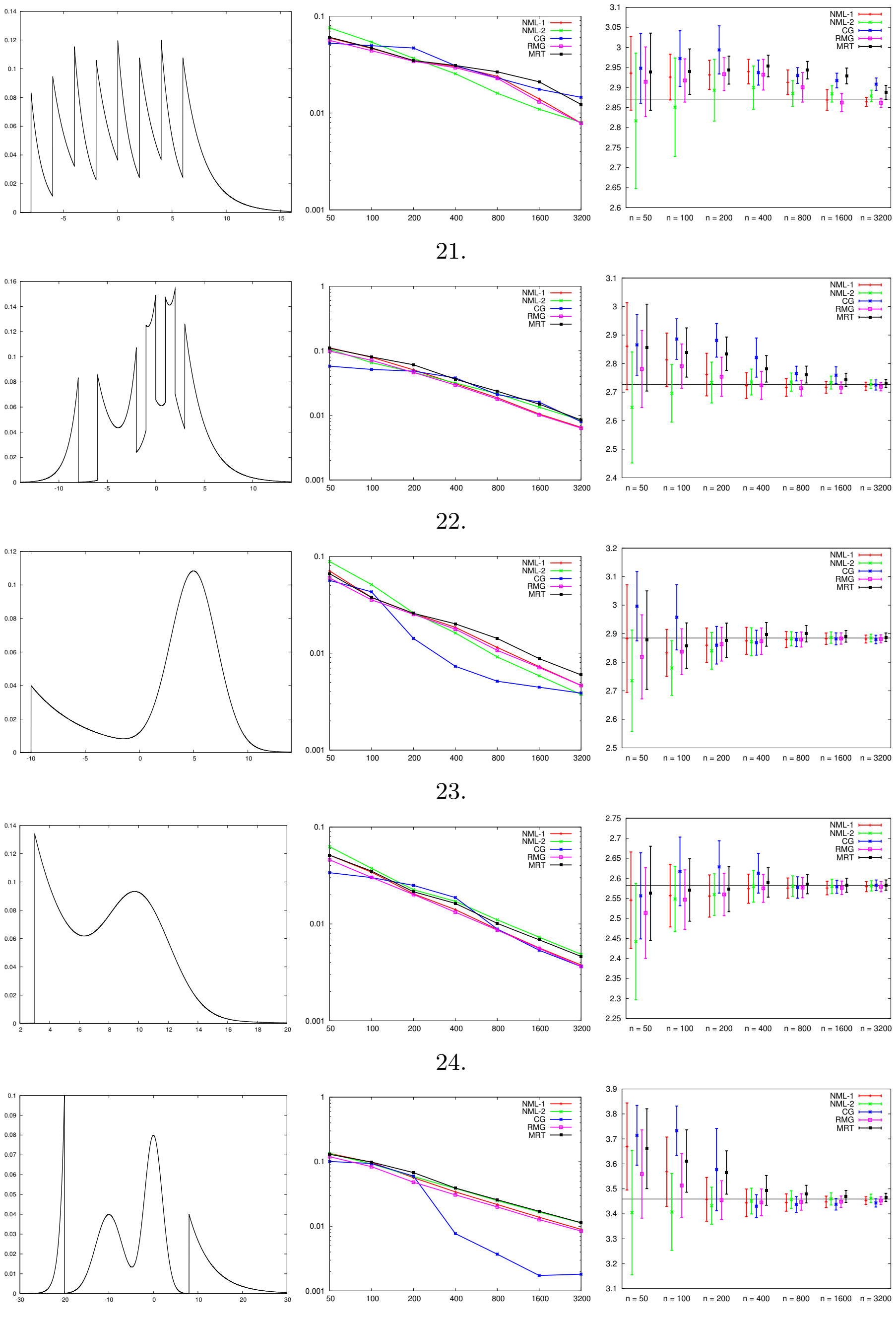
17.

18.

19.

20.

Figure A.6: Density and entropy estimation test results for densities 16–20. The colours for the methods: NML-1, NML-2, CG, RMG, MRT.



21.

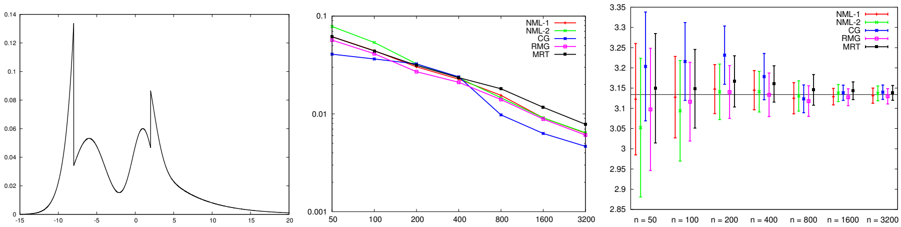
22.

23.

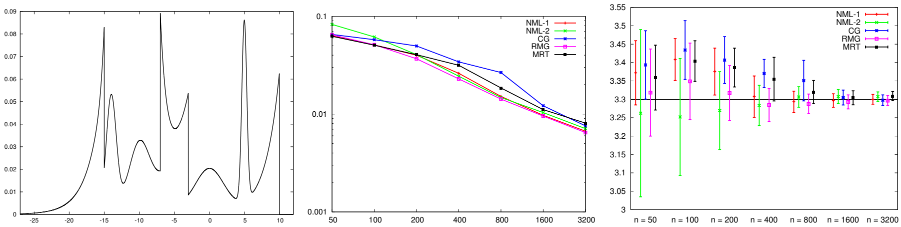
24.

25.

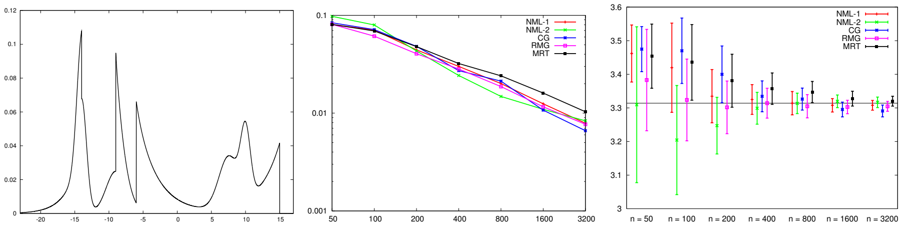
Figure A.7: Density and entropy estimation test results for densities 21–25. The colours for the methods: NML-1, NML-2, CG, RMG, MRT.



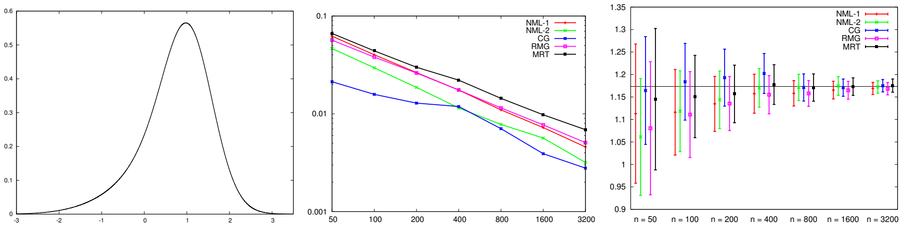
26.



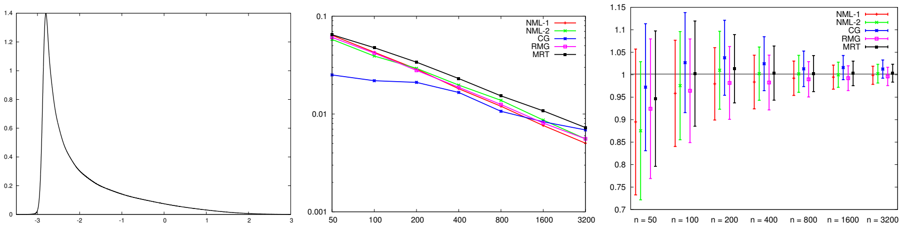
27.



28.

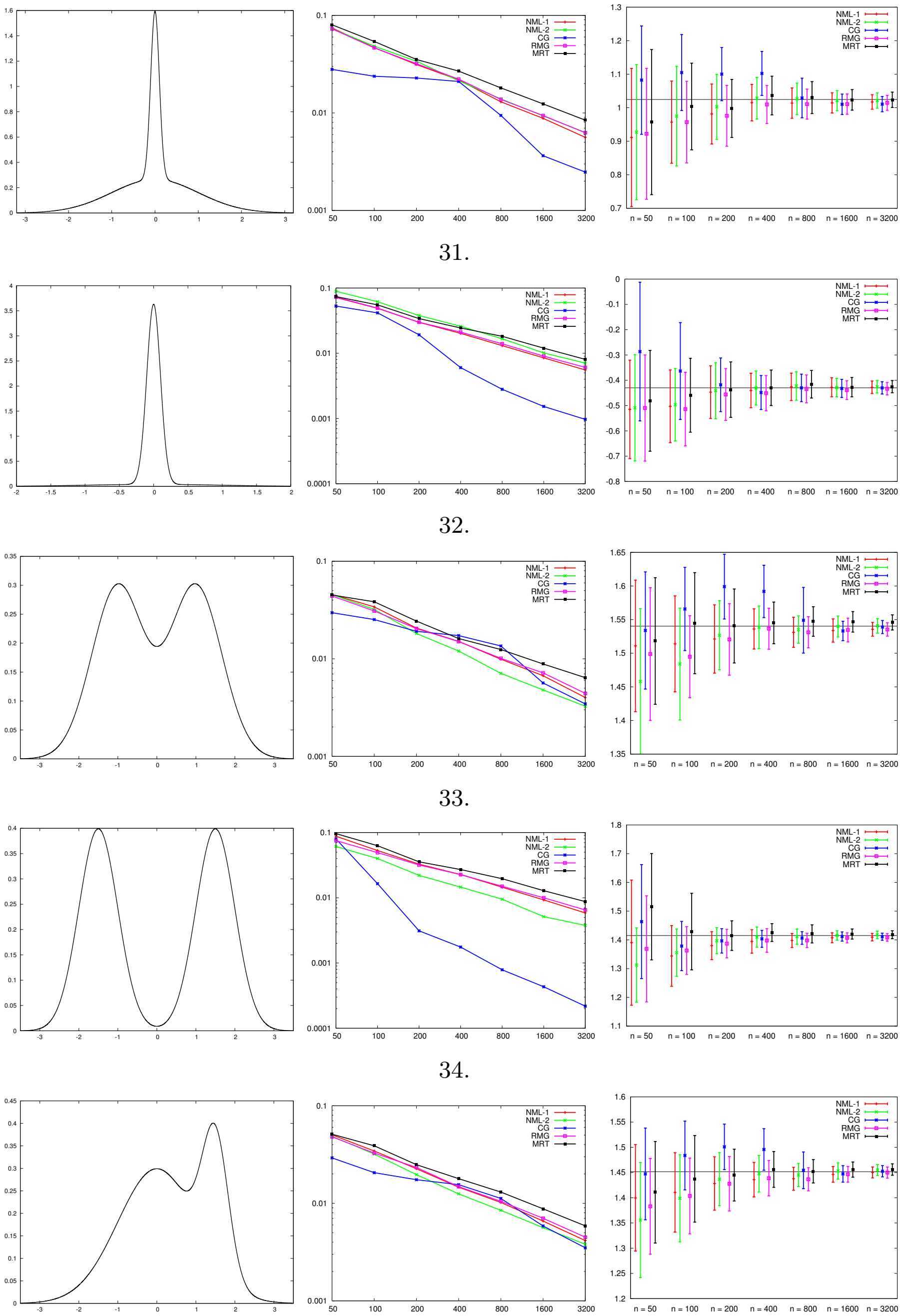


29.



30.

Figure A.8: Density and entropy estimation test results for densities 25–30. The colours for the methods: NML-1, NML-2, CG, RMG, MRT.



31.

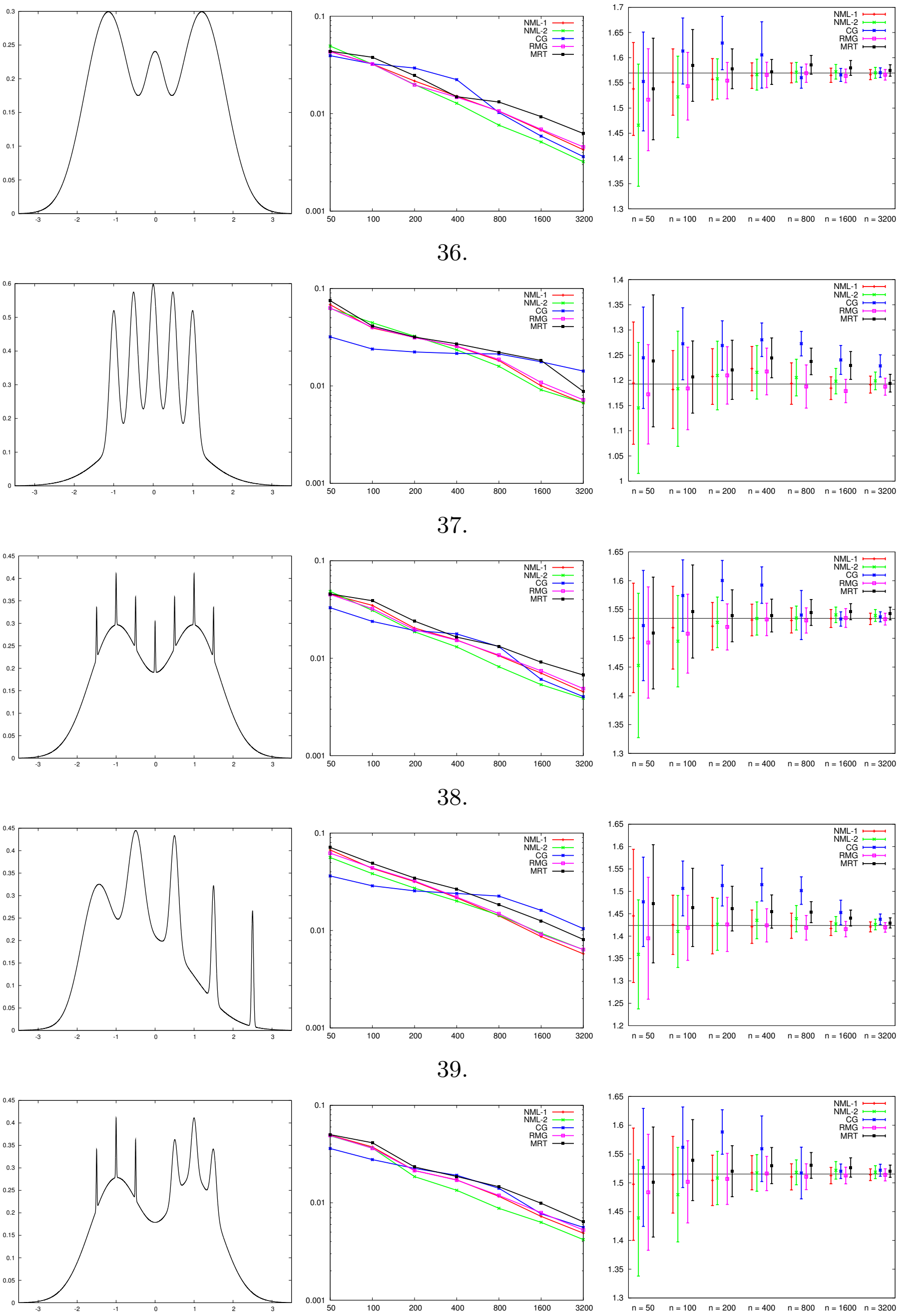
32.

33.

34.

35.

Figure A.9: Density and entropy estimation test results for densities 31–35. The colours for the methods: NML-1, NML-2, CG, RMG, MRT.



36.

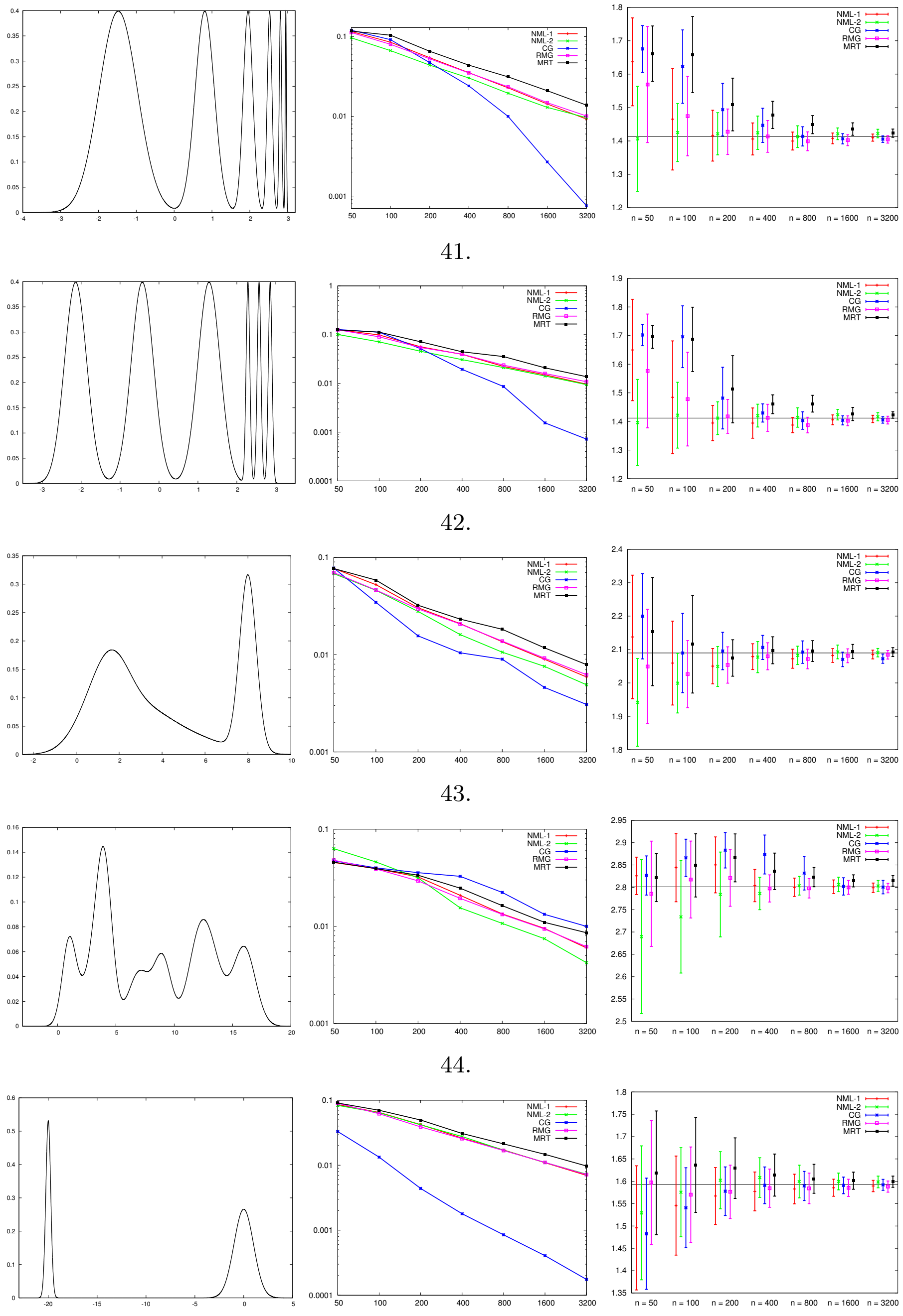
37.

38.

39.

40.

Figure A.10: Density and entropy estimation test results for densities 36–40. The colours for the methods: NML-1, NML-2, CG, RMG, MRT.



41.

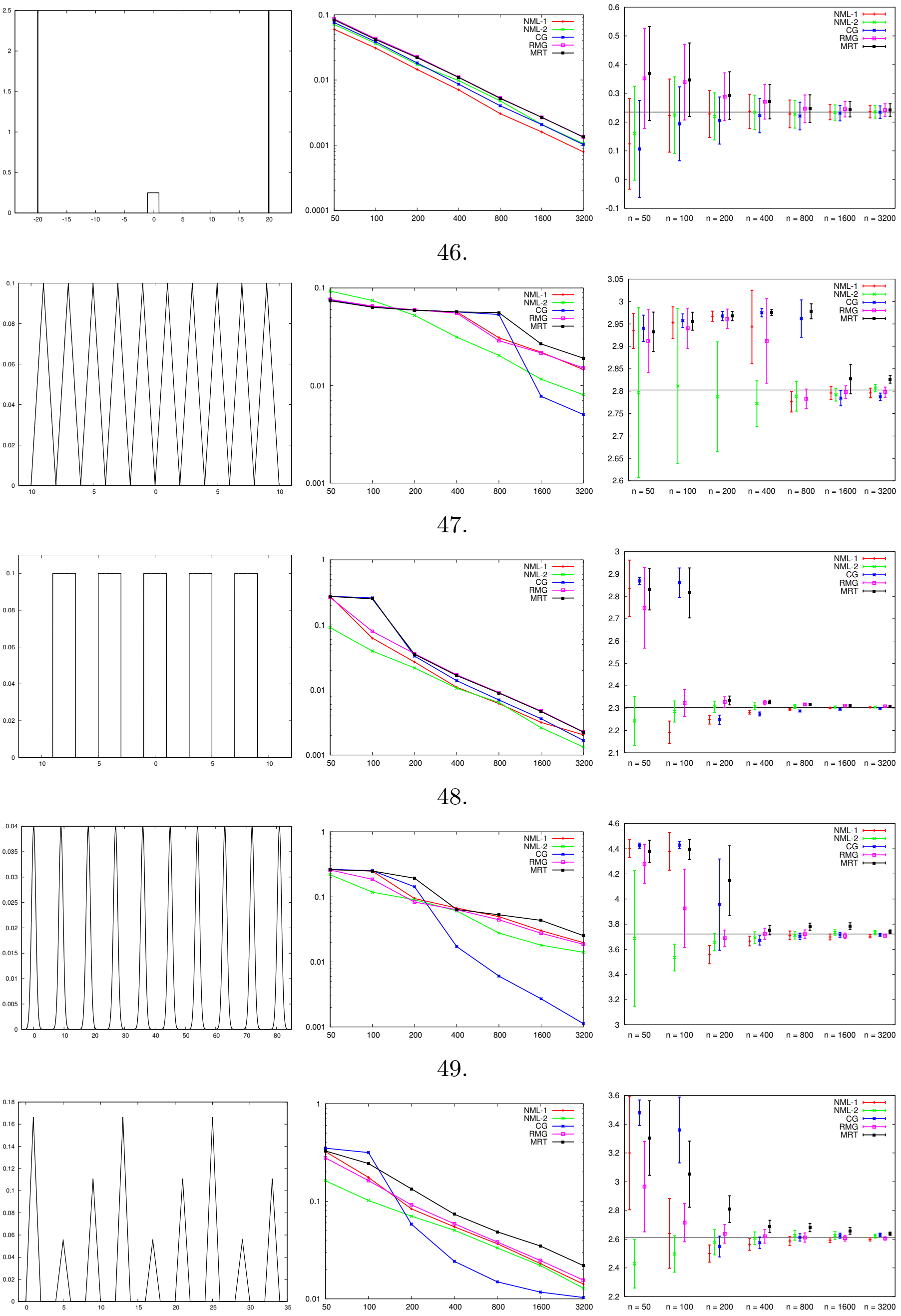
42.

43.

44.

45.

Figure A.11: Density and entropy estimation test results for densities 41–45. The colours for the methods: NML-1, NML-2, CG, RMG, MRT.



46.

47.

48.

49.

50.

Figure A.12: Density and entropy estimation test results for densities 46–50. The colours for the methods: NML-1, NML-2, CG, RMG, MRT.

Erratum

In Paper II at the end of Section 5, the definition of $f_{\mathbb{R}_+}$ in the integral is not correct. The calculation of the integral should be

$$\begin{aligned}
\int_0^{\epsilon} f_{\mathbb{R}_+}(x;\, b) &= \int_0^{\epsilon} \frac{1-\ln 2}{(\ln 2)^k} \frac{1}{1+\log\delta(\ln 2 - 1)} \frac{1}{(x+b)\,h(x+b)}\,\mathrm{d}x \\
&= \frac{1-\ln 2}{(\ln 2)^k} \frac{1}{1+\log\delta(\ln 2 - 1)} \int_b^{b+\epsilon} \frac{1}{y\,h(y)}\,\mathrm{d}y \\
&= \frac{1-\ln 2}{(\ln 2)^k} \frac{1}{1+\log\delta(\ln 2 - 1)} \Big/_{y=b}^{b+\epsilon} (\ln 2)^k \log^{(k)} y \\
&= \frac{(1-\ln 2)(\log\alpha - \log\delta)}{1+\log\delta(\ln 2 - 1)}\,,
\end{aligned}$$

and therefore

$$c = \left(1 - \frac{(1-\ln 2)(\log\alpha - \log\delta)}{1+\log\delta(\ln 2 - 1)} + \frac{f_{\mathbb{R}_+}(\epsilon;\, b)}{n-1}\epsilon\right)^{-1}.$$

The error does not affect the tests of the paper, because they use the parameter setting $b = 4$. It implies $k = 3$, $\delta = 1$, and c gets the correct value despite of the error.

References

[1] M. Abramson. Restricted combinations and compositions. *The Fibonacci Quarterly*, 14(5):439–452, 1976.

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

[3] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[4] A. Berlinet and L. Devroye. A comparison of kernel density estimates. Technical report, l'Institut de statistique de l'Université de Paris, 1994.

[5] L. Birgé and Y. Rozenholc. How many bins should be put in a regular histogram. *ESAIM : Probability and Statistics*, 10:24–45, February 2006.

[6] M. A. Carlton. *Applications of the Two-Parameter Poisson-Dirichlet Distribution.* PhD thesis, University of California, 1999.

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

[8] A. Dasgupta and A. E. Raftery. Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 98(441):294–302, March 1998.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[10] D. P. Foster and R. A. Stine. The contribution of parameters to stochastic complexity. In P. D. Grünwald, I. J. Myung, and M. A. Pitt, editors, *Advances in Minimum Description Length Theory and Applications*, pages 195–213. The MIT Press, 2005.

[11] P. D. Grünwald. *The Minimum Description Length Principle.* The MIT Press, 2007.

[12] S. Hirai and K. Yamanishi. Efficient computation of normalized maximum likelihood coding for Gaussian mixtures with its applications to optimal clustering. In *Proceedings of the 2011 IEEE International Symposium on Information Theory*, pages 1031–1035, 2011.

[13] S. Hirai and K. Yamanishi. Detecting changes of clustering structures using normalized maximum likelihood coding. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledege Discovery and Data Mining*, pages 343–351, 2012.

[14] A. J. Izenman. Recent developments in nonparametric density estimation. *Journal of the American Statistical Association*, 86(413):205–224, 1991.

[15] H. V. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. Sevcik, and T. Suel. Optimal histograms with quality guarantees. In A. Gupta, O. Shmueli, and J. Widom, editors, *Proceedings of the 24th International Conference on Very Large Data Bases*, pages 275–286, 1998.

[16] S. Kakade, M. Seeger, and D. Foster. Worst-case bounds for Gaussian process models. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 619–626. MIT Press, Cambridge, MA, 2006.

[17] P. Kontkanen and P. Myllymäki. A linear-time algorithm for computing the multinomial stochastic complexity. *Information Processing Letters*, 103(6):227–233, 2007.

[18] P. Kontkanen and P. Myllymäki. MDL histogram density estimation. In *Proceedings of the 11th International Workshop on Artificial Intelligence and Statistics*, March 2007.

[19] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri. An MDL framework for data clustering. In P. D. Grünwald, I. J. Myung, and M. A. Pitt, editors, *Advances in Minimum Description Length Theory and Applications*, pages 323–353. The MIT Press, 2005.

[20] G. Korodi and I. Tabus. An efficient normalized maximum likelihood algorithm for DNA sequence compression. *ACM Transactions on Information Systems*, 23(1):3–34, 2005.

[21] G. Korodi and I. Tabus. Normalized maximum likelihood model of order-1 for the compression of DNA sequences. In *Data Compression Conference*, pages 33–42, March 2007.

[22] G. Korodi, I. Tabus, J. Rissanen, and J. Astola. DNA sequence compression - based on the normalized maximum likelihood model. *Signal Processing Magazine, IEEE*, 24(1):47–53, January 2007.

[23] M. Li and P. Vitányi. *An introduction to Kolmogorov complexity and its applications.* Springer-Verlag, 2008.

[24] J. S. Marron and M. P. Wand. Exact mean integrated squared error. *The Annals of Statistics*, 20(2):712–736, 1992.

[25] G. Menez, M.-J. Rendas, and E. Thierry. Entropy estimation using MDL and piecewise constant density models. In *Proceedings of International Symposium on Information Theory and Its Applications*, 2008.

[26] D. Nicorici, O. Yli-Harja, and J. Astola. Stochastic complexity of vectors containing cluster structure. In P. D. Cristea, I. Tabus, and R. Tuduce, editors, *Proceedings of NSIP 2007 – International Workshop on Nonlinear Signal and Image Processing*, pages 164–169, 2007.

[27] J. Pitman. Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102:145–158, 1995.

[28] J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, 1989.

[29] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[30] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431, Jun 1983.

[31] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society*, 49(3):223–239 and 253–262, 1987.

[32] J. Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47, Jan 1996.

[33] J. Rissanen. MDL denoising. *IEEE Transactions on Information Theory*, 46(7):2537–2543, November 2000.

[34] J. Rissanen. Strong optimality of the normalized ML models as universal codes and information in data. *IEEE Transactions on Information Theory*, 47(5):1712–1717, 2001.

[35] J. Rissanen. *Information and Complexity in Statistical Modeling.* Springer Verlag, New York, 2007.

[36] J. Rissanen. *Optimal Estimation of Parameters.* Cambridge University Press, 2012.

[37] J. Rissanen, T. P. Speed, and B. Yu. Density estimation by stochastic complexity. *IEEE Transactions on Information Theory*, 38(2):315–323, 1992.

[38] T. Roos, P. Myllymaki, and J. Rissanen. MDL denoising revisited. *IEEE Transactions on Signal Processing*, 57(9):3347–3360, 2009.

[39] T. Roos, P. Myllymäki, and H. Tirri. On the behavior of MDL denoising. In R. Cowell and Z. Ghahramani, editors, *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 309–316, 2005.

[40] Y. Rozenholc, T. Mildenberger, and U. Gather. Combining regular and irregular histograms by penalized likelihood. *Computational Statistics and Data Analysis*, 54:3313–3323, 2010.

[41] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[42] D. W. Scott and S. R. Sain. Multidimensional density estimation. In C. Rao, E. Wegman, and J. Solka, editors, *Handbook of Statistics*, volume 24, pages 229–261. Elsevier, 2005.

[43] Y. M. Shtarkov. Universal sequential coding of single messages. *Problems of Information Transmission*, 23:3–17, 1987.

[44] R. J. Solomonoff. A formal theory of inductive inference, Part I. *Information and Control*, 7(1):1–22, 1964.

[45] R. J. Solomonoff. A formal theory of inductive inference, Part II. *Information and Control*, 7(2):224–254, 1964.

[46] I. Tabus, J. Rissanen, and J. Astola. Classification and feature gene selection using the normalized maximum likelihood model for discrete regression. *Signal Processing*, 83(4):713–727, 2003.

[47] I. Tabus, J. Rissanen, and J. Astola. Normalized maximum likelihood models for boolean regression with application to prediction and classification in genomics. In W. Zhang and I. Shmulevich, editors, *Computational and Statistical Approaches to Genomics*, pages 235–258. Springer US, 2006.

[48] C. Wallace and J. Patrick. Coding decision trees. *Machine Learning*, 11:7–22, 1993.