Available online at www.sciencedirect.com
ScienceDirect
Information Processing Letters 103 (2007) 227–233
Information Processing Letters
www.elsevier.com/locate/ipl

A linear-time algorithm for computing the multinomial stochastic complexity

Petri Kontkanen *, Petri Myllymäki

Complex Systems Computation Group (CoSCo), Helsinki Institute for Information Technology (HIIT), University of Helsinki, Finland and Helsinki University of Technology, P.O. Box 68 (Department of Computer Science), FIN-00014 University of Helsinki, Finland

Received 30 November 2006; received in revised form 14 February 2007; accepted 5 April 2007

Available online 20 April 2007

Communicated by P.M.B. Vitányi

Abstract

The minimum description length (MDL) principle is a theoretically well-founded, general framework for performing model class selection and other types of statistical inference. This framework can be applied for tasks such as data clustering, density estimation and image denoising. The MDL principle is formalized via the so-called normalized maximum likelihood (NML) distribution, which has several desirable theoretical properties. The codelength of a given sample of data under the NML distribution is called the stochastic complexity, which is the basis for MDL model class selection. Unfortunately, in the case of discrete data, straightforward computation of the stochastic complexity requires exponential time with respect to the sample size, since the definition involves an exponential sum over all the possible data samples of a fixed size. As a main contribution of this paper, we derive an elegant recursion formula which allows efficient computation of the stochastic complexity in the case of n observations of a single multinomial random variable with K values. The time complexity of the new method is $\mathcal{O}(n + K)$ as opposed to $\mathcal{O}(n \log n \log K)$ obtained with the previous results.

Keywords: Algorithms; Combinatorial problems; Computational complexity

1. Introduction

One of the most important problems in machine learning and statistics is *model class selection*, which is the task of selecting among a set of competing mathematical explanations the one that describes a given sample of data best. The *minimum description length* (MDL) principle developed in the series of papers [16–18] is a well-founded, general framework for performing model class selection and other types of statistical inference. The fundamental idea behind the MDL principle is that any regularity in data can be used to *compress* the data, i.e., to find a description or *code* of it such that this description uses less symbols than it takes to describe the data literally. The more regularities there are, the more the data can be compressed. According to the MDL principle, learning can be equated with finding regularities in data. Consequently, we can say that the more we are able to compress the data, the more we have learned about it.

As codes and probability distributions are inherently intertwined (see, e.g., [5]), an efficient code for a data

* Corresponding author.
E-mail address: petri.kontkanen@hiit.fi (P. Kontkanen).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2007.04.003

set can be regarded as a probabilistic model yielding a high probability (short codelength) to the data at hand. Considering all possible models is not computationally feasible, so in practice we have to restrict ourselves to some limited set of probabilistic models. Mathematically, a model class is defined as a set of probability distributions indexed by a parameter vector. A *universal model* assigns a probability distribution for fixed-size data samples given a model class in such a manner that the data is given a high probability whenever there exists a distribution in the model class that gives high probability to the data. In other words, a universal model represents (or mimics) the behavior of all the distributions in the model class. The model class selection task is then solved by choosing the model class for which the associated universal distribution assigns the highest probability to the observed data.

According to the MDL principle, the universal models are formalized via the normalized maximum likelihood (NML) distribution [23,18], and the corresponding codelength of a data sample under the NML distribution is called the *stochastic complexity* (SC). Consequently, MDL model class selection is based on minimization of the stochastic complexity.

The NML distribution has several theoretical optimality properties, which make it a very attractive candidate for performing model class selection and related tasks. It was originally [18,2] formulated as a unique solution to the minimax problem presented in [23], which implied that NML is the minimax optimal universal model. Later [19], it was shown that NML is also the minimax optimal universal model in the expectation sense. See Section 2 and [2,19,7,20] for more discussion on the theoretical properties of the NML.

On the practical side, NML has been successfully applied to several problems. We mention here some examples. First, in [14], NML was used for clustering of multi-dimensional data and its performance was compared to alternative approaches like Bayesian statistics. The results showed that the performance of NML was especially impressive with small sample sizes. Second, in [21], NML was applied to wavelet denoising of digital images. Since the MDL principle in general can be interpreted as separating information from noise, this approach is very natural. Third, a scheme for using NML for histogram density estimation was presented in [13]. In this work, the density estimation problem was regarded as a model class selection task. This approach allowed finding NML-optimal histograms with variable-width bins in a computationally efficient way, providing both the optimal number of bins and the location of the bin borders.

For multinomial (discrete) data, the definition of the NML distribution (and thus of the stochastic complexity) involves a normalizing sum over all the possible data samples of a fixed size. Unfortunately, in most cases, the computation of this normalizing sum is infeasible. The topic of this paper is the derivation of an efficient algorithm to calculate the stochastic complexity in the case of multinomial data with K possible values. The algorithm works in linear time with respect to the sample size n.

The problem of computing the multinomial stochastic complexity efficiently has been studied before. In [10], a quadratic-time algorithm was presented. This was later [9,12] improved to $\mathcal{O}(n \log n \log K)$. Although the exponentiality of the computation was removed by these algorithms, they are still superlinear with respect to the size of the data. Furthermore, the practical value of the $\mathcal{O}(n \log n \log K)$ algorithm is questionable due to numerical instability problems, while the linear-time algorithm presented in this paper can be easily implemented without such problems.

Several approximation schemes for computing the multinomial stochastic complexity have also been suggested. The accuracy of the approximations was studied empirically in [10], where it was observed that the error of the traditional *Bayesian Information Criterion* (BIC) [22] and *Rissanen's asymptotic expansion* [18] can be substantial, especially with small sample sizes or if the number of values K is large, while the *Szpankowski approximation* introduced in [10] was found to be very accurate. However, the task of computing the exact stochastic complexity has theoretical significance in itself. What is more, it is not clear how to extend the Szpankowski approximation beyond the multinomial case, while the exact computation methods can be directly applied in more complex cases, like the clustering model class discussed in [14]. Therefore, in the following we concentrate only on the exact computation of the stochastic complexity.

This paper is structured as follows. In Section 2 we discuss the basic properties of the MDL principle and the NML distribution. In Section 3 we instantiate the NML distribution for the multinomial model class. We will also shortly discuss the previous stochastic complexity computation algorithms. The topic of Section 4 is to derive the so-called regret generating function, which is then in Section 5 used as a basis for the new, linear-time algorithm. Finally, Section 6 gives some concluding remarks.

2. Properties of MDL and NML

The MDL principle has several desirable properties. Firstly, it automatically protects against overfitting in the model class selection process. Secondly, there is no need to assume that there exists some underlying "true" model, while most other statistical frameworks do. The model class is only used as a technical device for constructing an efficient code for describing the data. MDL is also closely related to Bayesian inference but there are some fundamental differences, the most important being that MDL is not dependent on any prior distribution, it only uses the data at hand. For more discussion on the theoretical motivations behind the MDL principle see, e.g., [18,2,26,19,7,20].

MDL model class selection is based on minimization of the stochastic complexity. In the following, we give the definition of the stochastic complexity and then proceed by discussing its theoretical properties.

Let $\mathbf{x}^n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be a data sample of n outcomes, where each outcome $\mathbf{x}_j$ is an element of some space of observations $\mathcal{X}$. The n-fold Cartesian product $\mathcal{X} \times \cdots \times \mathcal{X}$ is denoted by $\mathcal{X}^n$, so that $\mathbf{x}^n \in \mathcal{X}^n$. Consider a set $\Theta \subseteq \mathbb{R}^d$, where d is a positive integer. A class of parametric distributions indexed by the elements of Θ is called a *model class*. That is, a model class $\mathcal{M}$ is defined as

$$\mathcal{M} = \big\{ P(\cdot \mid \boldsymbol{\theta}) \colon \boldsymbol{\theta} \in \Theta \big\}. \tag{1}$$

Denote the maximum likelihood estimate of data $\mathbf{x}^n$ for a given model class $\mathcal{M}$ by $\hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M})$, i.e., $\hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}) = \arg\max_{\boldsymbol{\theta} \in \Theta} \{ P(\mathbf{x}^n \mid \boldsymbol{\theta}) \}$. The *normalized maximum likelihood* (NML) distribution [23] is now defined as

$$P_{\mathrm{NML}}(\mathbf{x}^n \mid \mathcal{M}) = \frac{P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}))}{\mathcal{C}(\mathcal{M}, n)}, \tag{2}$$

where the normalizing term $\mathcal{C}(\mathcal{M}, n)$ in the case of discrete data is given by

$$\mathcal{C}(\mathcal{M}, n) = \sum_{\mathbf{y}^n \in \mathcal{X}^n} P\big(\mathbf{y}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{y}^n, \mathcal{M})\big), \tag{3}$$

and the sum goes over the space of data samples of size n. If the data is continuous, the sum is replaced by the corresponding integral.

The stochastic complexity of the data $\mathbf{x}^n$ given a model class $\mathcal{M}$ is defined via the NML distribution as

$$\begin{aligned} \mathrm{SC}(\mathbf{x}^n \mid \mathcal{M}) &= -\log P_{\mathrm{NML}}(\mathbf{x}^n \mid \mathcal{M}) \\ &= -\log P\big(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M})\big) \\ &\quad + \log \mathcal{C}(\mathcal{M}, n), \end{aligned} \tag{4}$$

and the term $\log \mathcal{C}(\mathcal{M}, n)$ is called the *minimax regret* or *parametric complexity*. The minimax regret can be interpreted as measuring the logarithm of the number of essentially different (distinguishable) distributions in the model class. Intuitively, if two distributions assign high likelihood to the same data samples, they do not contribute much to the overall complexity of the model class, and the distributions should not be counted as different for the purposes of statistical inference. See [1] for more discussion on this topic.

The NML distribution (2) has several important theoretical optimality properties. The first one is that NML provides the unique solution to the minimax problem posed in [23],

$$\min_{\hat{P}} \max_{\mathbf{x}^n} \log \frac{P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}))}{\hat{P}(\mathbf{x}^n \mid \mathcal{M})}, \tag{5}$$

so that the minimizing $\hat{P}$ is the NML distribution, and the minimax regret

$$\log P\big(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M})\big) - \log \hat{P}(\mathbf{x}^n \mid \mathcal{M}) \tag{6}$$

is given by the parametric complexity $\log \mathcal{C}(\mathcal{M}, n)$. This means that the NML distribution is the *minimax optimal universal model* with respect to the model class $\mathcal{M}$, but note that the NML distribution itself typically does not belong to the model class.

A related property of NML involving expected regret was proven in [19]. This property states that NML also solves

$$\min_{\hat{P}} \max_{g} E_g \log \frac{P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}))}{\hat{P}(\mathbf{x}^n \mid \mathcal{M})}, \tag{7}$$

where the expectation is taken over $\mathbf{x}^n$ and g is the worst-case data generating distribution. The minimax expected regret is also given by $\log \mathcal{C}(\mathcal{M}, n)$.

3. NML for the multinomial model class

In the following, we will assume that our problem domain consists of a single discrete random variable X with K values, and that our data $\mathbf{x}^n = (x_1, \ldots, x_n)$ is multinomially distributed. Without loss of generality, the space of observations $\mathcal{X}$ can be assumed to be the set $\{1, 2, \ldots, K\}$. We denote the multinomial model classes by $\mathcal{M}_K$ and define

$$\mathcal{M}_K = \big\{ P(X \mid \boldsymbol{\theta}) \colon \boldsymbol{\theta} \in \Theta_K \big\}, \tag{8}$$

where Θ_K is the simplex-shaped parameter space

$$\Theta_K = \big\{ \boldsymbol{\theta} = (\theta_1, \ldots, \theta_K) \colon \theta_k \geqslant 0, \theta_1 + \cdots + \theta_K = 1 \big\}, \tag{9}$$

with $\theta_k = P(X = k \mid \boldsymbol{\theta})$, $k = 1, \ldots, K$.

It is well known (see, e.g., [10,14]) that the maximum likelihood parameters for the multinomial model class are given by $\hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}_K) = (h_1/n, \ldots, h_K/n)$, where h_k is the frequency (number of occurrences) of value k in $\mathbf{x}^n$. The NML distribution (2) for the model class $\mathcal{M}_K$ is then given by

$$P_{\mathrm{NML}}\big(\mathbf{x}^n \mid \mathcal{M}_K\big) = \frac{\prod_{k=1}^{K} (h_k/n)^{h_k}}{\mathcal{C}(\mathcal{M}_K, n)}, \tag{10}$$

where

$$\begin{aligned}\mathcal{C}(\mathcal{M}_K, n) &= \sum_{\mathbf{y}^n} P\big(\mathbf{y}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{y}^n, \mathcal{M}_K)\big) \\ &= \sum_{h_1+\cdots+h_K=n} \frac{n!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k}. \end{aligned} \tag{11}$$

In the following, we will simplify the notation by writing $\mathcal{C}(K, n)$ instead of $\mathcal{C}(\mathcal{M}_K, n)$.

It is clear that the maximum likelihood term in (10) can be computed in linear time by simply sweeping through the data once and counting the frequencies h_k. However, the normalizing sum $\mathcal{C}(K, n)$ (and thus also the parametric complexity $\log \mathcal{C}(K, n)$) involves a sum over an exponential (in K) number of terms. Consequently, the time complexity of computing the multinomial stochastic complexity is dominated by (11).

In [10,14] a recursion formula for removing the exponentiality of $\mathcal{C}(K, n)$ was presented. This formula is given by

$$\begin{aligned}\mathcal{C}(K_1 + K_2, n) = \sum_{r_1+r_2=n} &\frac{n!}{r_1! r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \\ &\cdot \mathcal{C}(K_1, r_1) \cdot \mathcal{C}(K_2, r_2), \end{aligned} \tag{12}$$

which holds for all $K_1, K_2 \geqslant 1$. A straightforward algorithm based on this formula was then used to compute $\mathcal{C}(K, n)$ in time $\mathcal{O}(n^2 \log K)$. See [10,14] for more details.

In [9,12] the quadratic-time algorithm was improved to $\mathcal{O}(n \log n \log K)$ by writing (11) as a convolution-type sum and then using the Fast Fourier Transform algorithm. However, the relevance of this result is unclear due to severe numerical instability problems it produces in practice.

Although the previous algorithms have succeeded in removing the exponentiality of the computation of the multinomial stochastic complexity, they are still superlinear with respect to n. In the next two sections we will derive a novel, linear-time algorithm for the problem.

4. The regret generating function

The mathematical technique of generating functions turns out to be the key element in the derivation of the new, efficient algorithm for computing the multinomial stochastic complexity. We start by reviewing some basic facts about generating functions.

One of the most powerful ways to analyze a sequence of numbers is to form a power series with the elements of the sequence as coefficients. The resulting function is called the *generating function* of the sequence. Generating functions can be seen as a bridge between discrete mathematics and continuous analysis. They can be used for, e.g., finding recurrence formulas and asymptotic expansions, proving combinatorial identities and finding statistical properties of a sequence. Good sources for further reading on generating functions are [25,6].

The (ordinary) generating function of a sequence $(a_n)_{n=0}^{\infty} = (a_0, a_1, a_2, \ldots)$ is defined as the series

$$A(z) = \sum_{n \geqslant 0} a_n z^n, \tag{13}$$

where z is a dummy symbol (or a complex variable). The importance of generating functions is that the function $A(z)$ is a compact representation of the whole sequence $(a_n)_{n=0}^{\infty}$. By studying this function we can get important information about the sequence, such as the exact or asymptotic form of the coefficients.

Our goal now is to find a computationally useful form for the generating function of the sequence

$$\big(\mathcal{C}(K, n)\big)_{n=0}^{\infty} = \big(\mathcal{C}(K, 0), \mathcal{C}(K, 1), \mathcal{C}(K, 2), \ldots\big). \tag{14}$$

A similar problem was studied in [24], and our derivation mostly follows it. Let us first consider the sequence $(n^n/n!)_{n=0}^{\infty}$. As in [24], we denote the function generating this sequence by $B(z)$. Squaring $B(z)$ yields

$$\begin{aligned} B^2(z) &= \left(\sum_{h_1 \geqslant 0} \frac{h_1^{h_1}}{h_1!} z^{h_1}\right) \cdot \left(\sum_{h_2 \geqslant 0} \frac{h_2^{h_2}}{h_2!} z^{h_2}\right) \\ &= \sum_{n \geqslant 0} \left(\sum_{h_1+h_2=n} \frac{n^n}{n!} \frac{n!}{h_1! h_2!} \frac{h_1^{h_1} h_2^{h_2}}{n^{h_1+h_2}}\right) z^n \\ &= \sum_{n \geqslant 0} \frac{n^n}{n!} \mathcal{C}(2, n) z^n. \end{aligned} \tag{15}$$

Thus, the function $B^2(z)$ generates the sequence $(\frac{n^n}{n!}\mathcal{C}(2, n))_{n=0}^{\infty}$. By basic combinatorics, it is straightforward to generalize this to

$$B^K(z) = \sum_{n\geqslant 0} \frac{n^n}{n!} \Bigg[\sum_{h_1+\cdots+h_K=n} \frac{n!}{h_1!\cdots h_K!} \cdot \prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k} \Bigg] z^n = \sum_{n\geqslant 0} \frac{n^n}{n!} \mathcal{C}(K,n) z^n, \tag{16}$$

which generates $(\frac{n^n}{n!}\mathcal{C}(K,n))_{n=0}^{\infty}$. The extra $n^n/n!$ term does not pose any problem, since it clearly can be canceled at the end of computation. Therefore this generating function can be used instead of the generating function of (14), and we call it the *regret generating function*.

However, there is no closed-form formula for $B(z)$ and little is known about the function in general. Therefore, we will write the function $B^K(z)$ in a different, more useful form using the so-called *Cayley's tree function* $T(z)$ [8,4], which generates the sequence $(n^{n-1}/n!)_{n=1}^{\infty}$:

$$T(z) = \sum_{n\geqslant 1} \frac{n^{n-1}}{n!} z^n. \tag{17}$$

This sequence counts the *rooted labeled trees* [3], hence the name of the function.

The connection between $T(z)$ and $B(z)$ is easy to derive (see [24]), and it is given by $B(z) = 1/(1-T(z))$. Consequently, the regret generating function can be written as

$$B^K(z) = \frac{1}{(1-T(z))^K}. \tag{18}$$

5. The linear-time algorithm

In this section, we will derive an elegant recurrence for the $\mathcal{C}(K,n)$ terms based on the regret generating function $B^K(z)$. At the end of the section, this recurrence is then used as a basis for the new, linear-time algorithm for computing the multinomial stochastic complexity.

We start by proving the following lemma:

Lemma 1. *For the tree function $T(z)$, it holds that*

$$zT'(z) = \frac{T(z)}{1-T(z)}. \tag{19}$$

Proof. A basic property of the tree function is the functional equation $T(z) = ze^{T(z)}$ (see, e.g., [8]). Differentiating this equation yields

$$T'(z) = e^{T(z)} + T(z)T'(z), \tag{20}$$

$$zT'(z)\big(1-T(z)\big) = ze^{T(z)}, \tag{21}$$

from which (19) follows. □

Now we can proceed to the main result of this paper:

Theorem 2. *The $\mathcal{C}(K,n)$ terms follow the recurrence*

$$\mathcal{C}(K+2,n) = \mathcal{C}(K+1,n) + \frac{n}{K} \cdot \mathcal{C}(K,n). \tag{22}$$

Proof. We start by multiplying and differentiating (16) as follows:

$$z \cdot \frac{d}{dz} \sum_{n\geqslant 0} \frac{n^n}{n!} \mathcal{C}(K,n) z^n = z \cdot \sum_{n\geqslant 1} n \cdot \frac{n^n}{n!} \mathcal{C}(K,n) z^{n-1} = \sum_{n\geqslant 0} n \cdot \frac{n^n}{n!} \mathcal{C}(K,n) z^n. \tag{23}$$

On the other hand, by manipulating (18) in the same way, we get

$$\begin{aligned} z \cdot \frac{d}{dz} \frac{1}{(1-T(z))^K} &= \frac{z \cdot K}{(1-T(z))^{K+1}} \cdot T'(z) \\ &= \frac{K}{(1-T(z))^{K+1}} \cdot \frac{T(z)}{1-T(z)} \qquad (24) \\ &= K\left(\frac{1}{(1-T(z))^{K+2}} - \frac{1}{(1-T(z))^{K+1}} \right) \\ &= K\Bigg(\sum_{n\geqslant 0} \frac{n^n}{n!} \mathcal{C}(K+2,n) z^n - \sum_{n\geqslant 0} \frac{n^n}{n!} \mathcal{C}(K+1,n) z^n \Bigg), \qquad (25) \end{aligned}$$

where (24) follows from Lemma 1. Comparing the coefficients of z^n in (23) and (25), we get

$$n \cdot \mathcal{C}(K,n) = K \cdot \big(\mathcal{C}(K+2,n) - \mathcal{C}(K+1,n) \big), \tag{26}$$

from which the theorem follows. □

An alternative proof of Theorem 2 is given in [11], where the so-called *tree polynomials* [8] are used. The proof given here, however, is shorter and more elegant.

It is now straightforward to write a linear-time algorithm for computing the multinomial stochastic complexity $\mathrm{SC}(\mathbf{x}^n \mid \mathcal{M}_K)$ based on Theorem 2. The process is described in Algorithm 1. The time complexity of the algorithm is clearly $\mathcal{O}(n+K)$, which is a major improvement over the previous methods. The algorithm is also very easy to implement and does not suffer from any numerical instability problems.

1: Count the frequencies $h_1, \ldots, h_K$ from the data $\mathbf{x}^n$
2: Compute the likelihood $P\big(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}_K)\big) = \prod_{k=1}^{K} \big(\frac{h_k}{n}\big)^{h_k}$
3: Set $\mathcal{C}(1, n) = 1$
4: Compute $\mathcal{C}(2, n) = \sum_{r_1+r_2=n} \frac{n!}{r_1! r_2!} \big(\frac{r_1}{n}\big)^{r_1} \big(\frac{r_2}{n}\big)^{r_2}$
5: **for** $k = 1$ to $K - 2$
6: Compute $\mathcal{C}(k+2, n) = \mathcal{C}(k+1, n) + \frac{n}{k} \cdot \mathcal{C}(k, n)$
7: **end for**
8: Output $\mathrm{SC}(\mathbf{x}^n \mid \mathcal{M}_K) = -\log P\big(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}_K)\big) + \log \mathcal{C}(K, n)$

Algorithm 1. The linear-time algorithm for computing $\mathrm{SC}(\mathbf{x}^n \mid \mathcal{M}_K)$.

6. Conclusion

In this paper we have derived a recursive formula for the exponential sums that appear in the definition of the normalized maximum likelihood distribution. Based on this formula, we presented the first linear-time algorithm for exact computation of the multinomial stochastic complexity. Besides being a theoretically important result, the new algorithm has also already been applied for efficient NML-optimal histogram density estimation in [13].

In the future, our plan is to extend the current work to more complex model classes such as Bayesian networks [15]. Even if it turns out that the regret generating function is not available in these cases, we believe that the current framework might still be useful in deriving accurate approximations of the stochastic complexity. Another natural area of future work is to apply the results of this paper to practical tasks such as classification.

Acknowledgements

The authors would like thank the anonymous reviewers for constructive comments. This work was supported in part by the Academy of Finland under the project Civi and by the Finnish Funding Agency for Technology and Innovation under the projects Kukot and PMMA. In addition, this work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

[1] V. Balasubramanian, MDL, Bayesian inference, and the geometry of the space of probability distributions, in: P. Grünwald, I.J. Myung, M. Pitt (Eds.), Advances in Minimum Description Length: Theory and Applications, MIT Press, 2006, pp. 81–98.
[2] A. Barron, J. Rissanen, B. Yu, The minimum description principle in coding and modeling, IEEE Transactions on Information Theory 44 (6) (1998) 2743–2760.
[3] C. Chauve, S. Dulucq, O. Guibert, Enumeration of some labelled trees, in: D. Krob, A.A. Mikhalev, A.V. Mikhalev (Eds.), Formal Power Series and Algebraic Combinatorics, FPSAC'00, Springer-Verlag, 2000, pp. 146–157.
[4] R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey, D.E. Knuth, On the Lambert W function, Advances in Computational Mathematics 5 (1996) 329–359.
[5] T. Cover, J. Thomas, Elements of Information Theory, John Wiley & Sons, New York, NY, 1991.
[6] R.L. Graham, D.E. Knuth, O. Patashnik, Concrete Mathematics, second ed., Addison-Wesley, 1994.
[7] P. Grünwald, Minimum description length tutorial, in: P. Grünwald, I.J. Myung, M. Pitt (Eds.), Advances in Minimum Description Length: Theory and Applications, MIT Press, 2006, pp. 23–79.
[8] D.E. Knuth, B. Pittel, A recurrence related to trees, Proceedings of the American Mathematical Society 105 (2) (1989) 335–349.
[9] M. Koivisto, Sum–product algorithms for the analysis of genetic risks, PhD thesis, Report A-2004-1, Department of Computer Science, University of Helsinki, 2004.
[10] P. Kontkanen, W. Buntine, P. Myllymäki, J. Rissanen, H. Tirri, Efficient computation of stochastic complexity, in: C. Bishop, B. Frey (Eds.), Proceedings of the Ninth International Conference on Artificial Intelligence and Statistics, Society for Artificial Intelligence and Statistics, 2003, pp. 233–238.
[11] P. Kontkanen, P. Myllymäki, Analyzing the stochastic complexity via tree polynomials, Technical Report 2005-4, Helsinki Institute for Information Technology (HIIT), 2005.
[12] P. Kontkanen, P. Myllymäki, A fast normalized maximum likelihood algorithm for multinomial data, in: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05), 2005.
[13] P. Kontkanen, P. Myllymäki, MDL histogram density estimation, in: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, San Juan, Puerto Rico, March 2007.
[14] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, H. Tirri, An MDL framework for data clustering, in: P. Grünwald, I.J. Myung, M. Pitt (Eds.), Advances in Minimum Description Length: Theory and Applications, MIT Press, 2006.
[15] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, San Mateo, CA, 1988.
[16] J. Rissanen, Modeling by shortest data description, Automatica 14 (1978) 445–471.
[17] J. Rissanen, Stochastic complexity, Journal of the Royal Statistical Society 49 (3) (1987) 223–239 and 252–265.
[18] J. Rissanen, Fisher information and stochastic complexity, IEEE Transactions on Information Theory 42 (1) (1996) 40–47.
[19] J. Rissanen, Strong optimality of the normalized ML models as universal codes and information in data, IEEE Transactions on Information Theory 47 (5) (2001) 1712–1717.
[20] J. Rissanen, Lectures on statistical modeling theory, August 2005. Available online at www.mdl-research.org.
[21] T. Roos, P. Myllymäki, H. Tirri, On the behavior of MDL denoising, in: Proceedings of the 10th International Workshop on

Artificial Intelligence and Statistics (AISTATS), 2005, pp. 309–316.

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

[23] Yu.M. Shtarkov, Universal sequential coding of single messages, Problems of Information Transmission 23 (1987) 3–17.

[24] W. Szpankowski, Average Case Analysis of Algorithms on Sequences, John Wiley & Sons, 2001.

[25] H.S. Wilf, generatingfunctionology, second ed., Academic Press, 1994.

[26] Q. Xie, A.R. Barron, Asymptotic minimax regret for data compression, gambling, and prediction, IEEE Transactions on Information Theory 46 (2) (2000) 431–445.