Department of Computer Science
Series of Publications A
Report A-2009-11

Computationally Efficient Methods for MDL-Optimal Density Estimation and Data Clustering

Petri Kontkanen

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium CK112, Exactum on November 30th, 2009, at 12 o'clock noon.

University of Helsinki
Finland

Contact information

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

Email address: postmaster@cs.Helsinki.FI (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 1911

Telefax: +358 9 191 51120

ISSN 1238-8645
ISBN 978-952-10-5900-1 (paperback)
ISBN 978-952-10-5901-8 (PDF)
Computing Reviews (1998) Classification: G.2.1, G.3, H.1.1
Helsinki 2009
Helsinki University Print

Computationally Efficient Methods for MDL-Optimal Density Estimation and Data Clustering

Petri Kontkanen

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
Petri.Kontkanen@cs.Helsinki.FI
http://www.cs.helsinki.fi/u/pkontkan/

PhD Thesis, Series of Publications A, Report A-2009-11
Helsinki, November 2009, 75+64 pages
ISSN 1238-8645
ISBN 978-952-10-5900-1 (paperback)
ISBN 978-952-10-5901-8 (PDF)

Abstract

The Minimum Description Length (MDL) principle is a general, well-founded theoretical formalization of statistical modeling. The most important notion of MDL is the stochastic complexity, which can be interpreted as the shortest description length of a given sample of data relative to a model class. The exact definition of the stochastic complexity has gone through several evolutionary steps. The latest instantation is based on the so-called Normalized Maximum Likelihood (NML) distribution which has been shown to possess several important theoretical properties. However, the applications of this modern version of the MDL have been quite rare because of computational complexity problems, i.e., for discrete data, the definition of NML involves an exponential sum, and in the case of continuous data, a multi-dimensional integral usually infeasible to evaluate or even approximate accurately. In this doctoral dissertation, we present mathematical techniques for computing NML efficiently for some model families involving discrete data. We also show how these techniques can be used to apply MDL in two practical applications: histogram density estimation and clustering of multi-dimensional data.

Computing Reviews (1998) Categories and Subject Descriptors:
G.2.1 [Combinatorics]
G.3 [Probability and Statistics]

H.1.1 [Models and Principles]: Systems and Information theory

General Terms:
statistics, machine learning, algorithms

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

Preface

This doctoral dissertation consists of an introductory part and six original research papers on the Minimum Description Length (MDL) principle. The focus of the papers is on the practical aspects of the MDL, not the theoretical properties of it. More precisely, the research papers present mathematical techniques that allow the efficient use of MDL in practical model class selection tasks. The papers also discuss how these techniques can be applied in real-world applications.

To give the reader preliminaries and motivation for easier understanding of the six research papers, the thesis starts with an introductory text. This part is intuitive in nature, all the technical details can be found in the respective research papers. The introductory part starts with a short review of the MDL principle and the NML distribution, which formally defines the MDL model class selection criterion (the stochastic complexity). Next, an overview of the mathematical techniques and algorithms for efficient computation of the NML is presented. These algorithms are then used in two practical applications: histogram density estimation and clustering of multi-dimensional data.

The final part of the introduction consists of two appendices. The first one provides the reader background to the mathematical tools used in various parts of the thesis. The topics of this appendix are complex analysis, formal power series, generating functions and asymptotic analysis of generating functions. Together these techniques provide a powerful toolbox for efficient NML computation for several interesting model families. The topic of the second appendix is the derivation of a novel, very accurate multinomial NML approximation. The derivation is based on the mathematical techniques described in the first appendix.

Acknowledgements

I would like to thank my advisor, Professor Petri Myllymäki, for several invaluable discussions and comments regarding this dissertation. I am also very grateful to Henry Tirri and Petri for providing me an inspiring and fun working environment in the CoSCo research group for so many years.

The Department of Computer Science of the University of Helsinki provided me a chance for a research visit to the University of Science and Technology, Hong Kong, where the introductory part of this thesis was written. I want to thank my host Professor Nevin Zhang and his research group for many useful discussions about various issues related to my research work. They also helped me a lot in adjusting to life and culture in Hong Kong.

In addition to the Department of Computer Science, the financial support from the Academy of Finland, the EU Network of Excellence PASCAL and the Finnish Funding Agency for Technology and Innovation (Tekes) have made it possible to conduct the research work of my dissertation.

I am also very grateful to my long time members of the CoSCo research group, Henry Tirri, Petri Myllymäki, Jussi Lahtinen, Tomi Silander, Tommi Mononen, Jukka Perkiö, Hannes Wettig, Antti Tuominen, Jukka Perkiö, Hannes Wettig, Kimmo Valtonen and Teemu Roos.

The pre-examiners of the manuscript of this dissertation were Professors Nevin Zhang and Samuel Kaski. I want to thank them for their effort and useful comments.

Finally, I want to thank my family and friends for all of their support and encouragement throughout the process of writing my dissertation.

Original publications and contributions

This doctoral dissertation is based on the following 6 research papers, which are referred in text as Papers I–VI.

Paper I: P. Kontkanen, W. Buntine, P. Myllymäki, J. Rissanen, and H. Tirri. Efficient computation of stochastic complexity. In C. Bishop and B. Frey, editors, *Proceedings of the Ninth International Conference on Artificial Intelligence and Statistics*, pages 233–238. Society for Artificial Intelligence and Statistics, 2003.

Paper II: P. Kontkanen and 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.

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

Paper IV: P. Kontkanen and P. Myllymäki. MDL histogram density estimation. In M. Meila and S. Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, March 2007.

Paper V: P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri. An MDL framework for data clustering. In P. Grünwald, I. Myung, and M. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*. The MIT Press, 2005.

Paper VI: P. Kontkanen and P. Myllymäki. An empirical comparison of NML clustering algorithms. In *Proceedings of the 2008 International Conference on Information Theory and Statistical Learning (ITSL-08)*, 2008.

The papers are re-printed at the end of the thesis.

In Papers I-III we develop algorithms for efficient computation of the NML in the case of the multinomial and Naive Bayes model family. The topic of Papers IV–VI is to show how NML can be applied to practical problems. The main contributions and short descriptions of the six papers are listed here:

Paper I: We introduce the first polynomial-time algorithm for computing the stochastic complexity (NML) for the multinomial and Naive Bayes model families. The running time of the algorithm is quadratic with respect to the sample size. We also present three stochastic complexity approximation algorithms and study their accuracy empirically.

Paper II: We improve the time complexity of the algorithm presented in Paper I to $\mathcal{O}(n \log n)$, where n is the sample size. The new algorithm is based on the convolution theorem and the Fast Fourier Transform (FFT) algorithm.

Paper III: We derive a recursion formula that can be used straightforwardly to compute the multinomial stochastic complexity in linear time. The mathematical technique applied here is generating functions.

Paper IV: We regard histogram density estimation as a model class selection problem and apply the minimum description length (MDL) principle to it. Using the results from Paper III, we show how to efficiently compute the stochastic complexity for the histogram densities. Furthermore, we derive a dynamic programming algorithm that can be used to find the globally optimal histogram in polynomial time.

Paper V: Clustering is one of the central concepts in the field of unsupervised data analysis. We regard clustering as a problem of partitioning the data into mutually exclusive clusters so that similar data vectors are grouped together. The number of clusters is unknown, and determining the optimal number is part of the clustering problem. For solving this problem, we suggest an information-theoretic framework based on the MDL principle. For computing the NML for the clustering model class, we use the algorithms of Papers I and II.

Paper VI: We compare empirically various algorithms for finding candidate solutions to the clustering problem discussed in Paper V. We present five algorithms for the task and use several real-world data sets to test the

algorithms. The results show that the traditional EM and K-means algorithms perform poorly. Furthermore, our novel hybrid clustering algorithm turns out to produce the best results.

In all six papers, the contribution of the current author is significant. In Paper I, the quadratic-time algorithm for the multinomial model family is due to Wray Buntine. The idea of applying MDL to the clustering problem in Paper V is by Petri Myllymäki.

Contents

Chapter 1

Introduction

Many problems in science can be cast as *model class selection* tasks, i.e., as tasks 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 [53, 54, 56] 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 them.

The MDL principle has several desirable properties. Firstly, it automatically protects against overfitting in the model class selection process. Secondly, this statistical framework does not – unlike most other frameworks – assume that there exists some underlying "true" model. 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 the Bayesian inference but there are some fundamental differences, the most important being that MDL does not need any prior distribution; it only uses the data at hand. For more discussion on the theoretical motivations behind the MDL principle see, e.g., [56, 5, 72, 57, 21, 58].

MDL model class selection is based on a quantity called *stochastic complexity*, which is the shortest description length of a given data relative to a model class. The stochastic complexity is defined via the normalized maximum likelihood (NML) distribution [63, 56]. For multinomial (discrete) data, this definition involves a normalizing sum over all the possible data

samples of a fixed size. The logarithm of this sum is called the *parametric complexity* or *regret*, which can be interpreted as the amount of complexity of the model class. If the data is continuous, the sum is replaced by the corresponding integral.

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 [56, 5] formulated as the unique solution to a minimax problem presented in [63], which implied that NML is the minimax optimal universal model. Later [57], it was shown that NML is also the solution to a related problem involving expected regret. See Section 2.2 and [5, 57, 21, 58] for more discussion on the theoretical properties of the NML.

Many scientific problems involve large data sets. In order to apply NML for these tasks one needs to develop suitable NML computation methods since the normalizing sum or integral in the definition of the NML is typically difficult to compute directly. The introductory part of this thesis starts by presenting algorithms for efficient computation of the NML for both one- and multi-dimensional discrete data. The model families used here are the multinomial and the Naive Bayes, and the discussion is based on the Papers I–III. In the multinomial case, the most efficient algorithm based on the technique of generating functions is linear with respect to the sample size, while the Naive Bayes algorithm is quadratic.

The task of finding efficient NML computation algorithms is a relatively new topic, and there are only few related studies. In [50], NML for the multinomial model family was written in another form, which resulted in another linear-time algorithm. The same paper also studied the connection between the multinomial NML and the so-called *birthday problem* [15], which is a classical problem of probability theory. A study of how the multinomial NML can be computed in sub-linear time with a finite precision is presented in [47]. The algorithm has time complexity $\mathcal{O}(\sqrt{dn})$, where d is the precision in digits and n is the sample size. In [49], new theoretically interesting recurrence formulas for NML computation are derived. A new quadratic-time algorithm for computing the parametric complexity in the case of Naive Bayes is presented in [46]. This algorithm is based on the so-called *Miller formula* [25] for computing the powers of formal power series.

There has also been studies on computing NML for more complex model families. In [70, 42, 48], algorithms for so-called *Bayesian forests* are presented. However, these algorithms are exponential with respect to the number of values of the domain variables. One solution to this problem

is suggested in [61], where the NML criterion is modified to a computationally less demanding form called the *factorized NML*. Initial empirical results show that this new criterion can be useful in model class selection problems.

The second part of the thesis describes how NML can be applied to practical problems using the techniques of the first part. Due to the computational efficiency problems, there are relatively few applications of NML. However, the existing applications demonstrate that NML works very well in practice and provides in many cases superior results when compared to alternative approaches. The first application discussed in the thesis is the NML-optimal histogram density estimation suggested in Paper IV. This framework provides both the optimal number of bins and the location of the bin borders of the histogram in polynomial time. The second application is the NML clustering of multi-dimensional discrete data introduced in Paper V. The optimization aspect of the clustering problem was studied in Paper VI, where five algorithms for efficiently searching the exponentially-sized clustering space were compared. See Chapter 4 for related work and more discussion on NML applications in general.

This thesis is structured as follows. In Chapter 2 we discuss the basic properties of the MDL principle and the NML distribution. We also instantiate NML for the two model families. In Chapter 3 we present both exact and approximative computation algorithms for NML. The chapter also includes an empirical comparison of three NML approximations for the multinomial model family. The topic of Chapter 4 is to show how NML can be applied in two practical tasks: density estimation and data clustering. Chapter 5 gives some concluding remarks and ideas for future work. The thesis then continues with two appendices: Appendix A provides mathematical background to the techniques used in the thesis and Appendix B gives a full derivation of the accurate multinomial NML approximation called the Szpankowski approximation. Finally, the six original research papers are re-printed at the end of the thesis.

Chapter 2

Stochastic Complexity

The MDL model class selection is based on minimization of the stochastic complexity. In the following, we first define the model class selection problem. Then we proceed by giving the definition of the stochastic complexity based on the normalized maximum likehood distribution and discuss its theoretical properties. Finally, we instantiate the NML for the multinomial and Naive Bayes model families.

2.1 Model Classes and Families

Let $\mathbf{x}^n = (x_1, \ldots, x_n)$ be a data sample of n outcomes, where each outcome 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

$$\mathcal{M} = \{P(\cdot \mid \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}, \tag{2.1}$$

and the set Θ is called the *parameter space*.

Consider now a set $\Phi \subseteq \mathbb{R}^e$, where e is a positive integer. Define a set $\mathcal{F}$ by

$$\mathcal{F} = \{\mathcal{M}(\boldsymbol{\phi}) : \boldsymbol{\phi} \in \Phi\}. \tag{2.2}$$

The set $\mathcal{F}$ is called a *model family*, and each of the elements $\mathcal{M}(\boldsymbol{\phi})$ is a model class. The associated parameter space is denoted by $\Theta_{\boldsymbol{\phi}}$. The model class selection problem can now be defined as a process of finding the parameter vector $\boldsymbol{\phi}$, which is optimal according to some pre-determined criteria. In Sections 2.3 – 2.4 we discuss two specific model families, which will make these definitions more concrete.

2.2 The Normalized Maximum Likelihood (NML) Distribution

One of the most theoretically and intuitively appealing model class selection criteria is the *stochastic complexity*. Denote first the maximum likelihood estimate of data $\mathbf{x}^n$ for a given model class $\mathcal{M}(\boldsymbol{\phi})$ by $\hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}(\boldsymbol{\phi}))$, i.e., $\hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}(\boldsymbol{\phi})) = \arg\max_{\boldsymbol{\theta} \in \Theta_{\boldsymbol{\phi}}} \{P(\mathbf{x}^n \mid \boldsymbol{\theta})\}$. The *normalized maximum likelihood* (NML) distribution [63] is now defined as

$$P_{\mathrm{NML}}(\mathbf{x}^n \mid \mathcal{M}(\boldsymbol{\phi})) = \frac{P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}(\boldsymbol{\phi})))}{\mathcal{C}(\mathcal{M}(\boldsymbol{\phi}), n)}, \tag{2.3}$$

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

$$\mathcal{C}(\mathcal{M}(\boldsymbol{\phi}), n) = \sum_{\mathbf{y}^n \in \mathcal{X}^n} P(\mathbf{y}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{y}^n, \mathcal{M}(\boldsymbol{\phi}))), \tag{2.4}$$

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 $\mathcal{M}(\boldsymbol{\phi})$, is defined via the NML distribution as

$$\mathrm{SC}(\mathbf{x}^n \mid \mathcal{M}(\boldsymbol{\phi})) = -\log P_{\mathrm{NML}}(\mathbf{x}^n \mid \mathcal{M}(\boldsymbol{\phi})) \tag{2.5}$$

$$= -\log P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}(\boldsymbol{\phi}))) + \log \mathcal{C}(\mathcal{M}(\boldsymbol{\phi}), n), \tag{2.6}$$

and the term $\log \mathcal{C}(\mathcal{M}(\boldsymbol{\phi}), n)$ is called the *(minimax) regret* or *parametric complexity*. The 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 [4] for more discussion on this topic.

The NML distribution (2.3) has several important theoretical optimality properties. The most important one is that NML provides a unique solution to the minimax problem posed in [63]:

$$\min_{\hat{P}} \max_{\mathbf{x}^n} \log \frac{P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}(\boldsymbol{\phi})))}{\hat{P}(\mathbf{x}^n \mid \mathcal{M}(\boldsymbol{\phi}))}, \tag{2.7}$$

where $\hat{P}$ can be any distribution over the data $\mathbf{x}^n$. The minimizing $\hat{P}$ is the NML distribution, and the minimax regret

$$\log P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}(\boldsymbol{\phi}))) - \log \hat{P}(\mathbf{x}^n \mid \mathcal{M}(\boldsymbol{\phi})) \tag{2.8}$$

is given by the parametric complexity $\log \mathcal{C}(\mathcal{M}(\boldsymbol{\phi}), n)$. This means that the NML distribution is the *minimax optimal universal model*. The term universal model in this context means that the NML distribution represents (or mimics) the behaviour of all the distributions in the model class $\mathcal{M}(\boldsymbol{\phi})$. Note that the NML distribution itself does not have to belong to the model class, and typically it does not. For more discussion on the theoretical properties of NML, see [5, 57, 21, 58].

2.3 NML for the Multinomial Model Family

In the case of discrete data, the simplest model family is the *multinomial*. The data is assumed to be one-dimensional and have only a finite set of possible values. Although simple, the multinomial model family has practical applications. In Paper IV, multinomial NML was used for histogram density estimation, and the problem was regarded as a model class selection task. The NML-optimal histograms were later [12] used as attribute models for Naive Bayes classifier.

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. The space of observations $\mathcal{X}$ is now the set $\{1, 2, \ldots, K\}$. The corresponding model family $\mathcal{F}_{\text{MN}}$ is defined by

$$\mathcal{F}_{\text{MN}} = \{\mathcal{M}(\boldsymbol{\phi}) : \boldsymbol{\phi} \in \Phi_{\text{MN}}\}, \tag{2.9}$$

where $\Phi_{\text{MN}} = \{1, 2, 3, \ldots\}$. Since the parameter vector $\boldsymbol{\phi}$ is in this case a single integer K, we denote the multinomial model classes by $\mathcal{M}(K)$ for simplicity and define

$$\mathcal{M}(K) = \{P(\cdot \mid \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta_K\}, \tag{2.10}$$

where Θ_K is the simplex-shaped parameter space

$$\Theta_K = \{(\pi_1, \ldots, \pi_K) : \pi_k \geq 0,\ \pi_1 + \cdots + \pi_K = 1\}, \tag{2.11}$$

with $\pi_k = P(X = k),\ k = 1, \ldots, K$.

Assume the data points x_j are independent and identically distributed (i.i.d.). The NML distribution (2.3) for the model class $\mathcal{M}(K)$ is now given by (see Papers I and V)

$$P_{\text{NML}}(\mathbf{x}^n \mid \mathcal{M}(K)) = \frac{\prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k}}{\mathcal{C}(\mathcal{M}(K), n)}, \tag{2.12}$$

where h_k is the frequency (number of occurrences) of value k in $\mathbf{x}^n$, and

$$\mathcal{C}(\mathcal{M}(K), n) = \sum_{\mathbf{y}^n} P(\mathbf{y}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{y}^n, \mathcal{M}(K))) \tag{2.13}$$

$$= \sum_{h_1+\cdots+h_K=n} \frac{n!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k}. \tag{2.14}$$

2.4 NML for the Naive Bayes Model Family

The one-dimensional case discussed in the previous section is not adequate for many real-world situations, where data are typically multi-dimensional, involving complex dependencies between the domain variables. In Paper I a quadratic-time algorithm for computing the NML for a specific multivariate model family, usually called the Naive Bayes, was derived. This model family has been very successful in practice in mixture modeling [41], clustering of data (Paper V), case-based reasoning [39], classification [22, 40] and data visualization [33].

Let us assume that our problem domain consists of m primary variables $X_1, \ldots, X_m$ and a special variable X_0, which can be one of the variables in our original problem domain or it can be latent. Assume that the variable X_i has K_i values and that the extra variable X_0 has K_0 values. The data $\mathbf{x}^n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ consist of observations of the form $\mathbf{x}_j = (x_{j0}, x_{j1}, \ldots, x_{jm}) \in \mathcal{X}$, where

$$\mathcal{X} = \{1, 2, \ldots, K_0\} \times \{1, 2, \ldots, K_1\} \times \cdots \times \{1, 2, \ldots, K_m\}. \tag{2.15}$$

The Naive Bayes model family $\mathcal{F}_{\mathrm{NB}}$ is defined by

$$\mathcal{F}_{\mathrm{NB}} = \{\mathcal{M}(\boldsymbol{\phi}) : \boldsymbol{\phi} \in \Phi_{\mathrm{NB}}\} \tag{2.16}$$

with $\Phi_{\mathrm{NB}} = \{1, 2, 3, \ldots\}^{m+1}$. The corresponding model classes are denoted by $\mathcal{M}(K_0, K_1, \ldots, K_m)$:

$$\mathcal{M}(K_0, K_1, \ldots, K_m) = \{P_{\mathrm{NB}}(\cdot \mid \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta_{K_0, K_1, \ldots, K_m}\}. \tag{2.17}$$

The basic Naive Bayes assumption is that given the value of the special variable, the primary variables are independent. We have consequently

$$P_{\mathrm{NB}}(X_0 = x_0, X_1 = x_1, \ldots, X_m = x_m \mid \boldsymbol{\theta}) = P(X_0 = x_0 \mid \boldsymbol{\theta}) \cdot \prod_{i=1}^{m} P(X_i = x_i \mid X_0 = x_0, \boldsymbol{\theta}). \tag{2.18}$$

Furthermore, we assume that the distribution of $P(X_0 \mid \boldsymbol{\theta})$ is multinomial with parameters $(\pi_1, \ldots, \pi_{K_0})$, and each $P(X_i \mid X_0 = k, \boldsymbol{\theta})$ is multinomial with parameters $(\sigma_{ik1}, \ldots, \sigma_{ikK_i})$. The whole parameter space is then

$$\begin{aligned}\Theta_{K_0,K_1,\ldots,K_m} =&\{(\pi_1, \ldots, \pi_{K_0}), (\sigma_{111}, \ldots, \sigma_{11K_1}), \ldots, (\sigma_{mK_01}, \ldots, \sigma_{mK_0K_m}) : \\ &\pi_k \geq 0,\ \sigma_{ikl} \geq 0,\ \pi_1 + \cdots + \pi_{K_0} = 1,\ \sigma_{ik1} + \cdots + \sigma_{ikK_i} = 1, \\ &i = 1, \ldots, m,\ k = 1, \ldots K_0\}, \end{aligned} \tag{2.19}$$

and the parameters have interpretations $\pi_k = P(X_0 = k)$ and $\sigma_{ikl} = P(X_i = l \mid X_0 = k)$.

Assuming i.i.d., the NML distribution for the Naive Bayes can now be written as (see Paper V)

$$P_{\mathrm{NML}}(\mathbf{x}^n \mid \mathcal{M}(K_0, K_1, \ldots, K_m)) = \frac{\prod_{k=1}^{K_0} \left(\frac{h_k}{n}\right)^{h_k} \prod_{i=1}^{m} \prod_{l=1}^{K_i} \left(\frac{f_{ikl}}{h_k}\right)^{f_{ikl}}}{\mathcal{C}(\mathcal{M}(K_0, K_1, \ldots, K_m), n)}, \tag{2.20}$$

where h_k is the number of times X_0 has value k in $\mathbf{x}^n$, f_{ikl} is the number of times X_i has value l when the special variable has value k, and $\mathcal{C}(\mathcal{M}(K_0, K_1, \ldots, K_m), n)$ is given by

$$\begin{aligned}&\mathcal{C}(\mathcal{M}(K_0, K_1, \ldots, K_m), n) \\ &\qquad = \sum_{h_1 + \cdots + h_{K_0} = n} \frac{n!}{h_1! \cdots h_{K_0}!} \prod_{k=1}^{K_0} \left(\frac{h_k}{n}\right)^{h_k} \prod_{i=1}^{m} \mathcal{C}(\mathcal{M}(K_i), h_k). \end{aligned} \tag{2.21}$$

Chapter 3

Efficient Computation of NML

In the previous chapter we saw that in the case of discrete data the definition of the NML distribution involves a sum over all the possible data samples of fixed size. Direct computation of this sum takes exponential time even in the case of a simple multinomial model. In this chapter we present efficient algorithms for computing this sum for two model families, the multinomial and Naive Bayes. For interesting algorithms for computing the NML for a more complex model family called the *Bayesian forests*, see [70, 42, 48].

3.1 The Multinomial Model Family

3.1.1 Exact Computation Algorithms

In the previous chapter we saw that the NML distribution for the multinomial model family (2.12) consists of two parts: the likelihood and the parametric complexity (2.14). It is clear that the likelihood term can be computed in linear time by simply sweeping through the data once and counting the frequencies h_k. However, the normalizing sum $\mathcal{C}(\mathcal{M}(K), n)$ (and thus also the parametric complexity $\log \mathcal{C}(\mathcal{M}(K), n)$) involves a sum over an exponential number of terms. Consequently, the time complexity of computing the multinomial NML is dominated by (2.14).

In Paper I, a recursion formula for removing the exponentiality of $\mathcal{C}(\mathcal{M}(K), n)$ was presented. This formula is given by

$$\mathcal{C}(\mathcal{M}(K), n) = \sum_{r_1+r_2=0}^{n} \frac{n!}{r_1! r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \cdot \mathcal{C}(\mathcal{M}(K^*), r_1) \cdot \mathcal{C}(\mathcal{M}(K - K^*), r_2), \quad (3.1)$$

which holds for all $K^* = 1, \ldots, K-1$. A straightforward algorithm based on this formula was then used to compute $\mathcal{C}(\mathcal{M}(K), n)$ in time $\mathcal{O}\left(n^2 \log K\right)$. See Papers I and V for more details.

In Paper II (see also [31]), the quadratic-time algorithm was improved to $\mathcal{O}\left(n \log n \log K\right)$ by writing (3.1) 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 easily produces in practice. See Paper II for more details.

Although the algorithms described above have succeeded in removing the exponentiality of the computation of the multinomial NML, they are still superlinear with respect to n. In Paper III the first linear-time algorithm based on the mathematical technique of generating functions was derived for the problem. The algorithm is based on the following theorem:

Theorem 3.1 *The $\mathcal{C}(\mathcal{M}(K), n)$ terms satisfy the recurrence*

$$\mathcal{C}(\mathcal{M}(K+2), n) = \mathcal{C}(\mathcal{M}(K+1), n) + \frac{n}{K} \cdot \mathcal{C}(\mathcal{M}(K), n). \tag{3.2}$$

Proof. See Paper III. □

It is now straightforward to write a linear-time algorithm for computing the multinomial NML $P_{\mathrm{NML}}(\mathbf{x}^n \mid \mathcal{M}(K))$ based on Theorem 3.1. The process is described in Algorithm 1. The time complexity of the algorithm is

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

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

clearly $\mathcal{O}\left(n + K\right)$, 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. See Paper III for more discussion of the algorithm.

3.1.2 NML Approximations

In the previous section we presented exact NML computation algorithms for multinomial data. The time complexity of the most efficient method was shown to be linear with respect to the size of the data, which can sometimes be too slow for demanding tasks. Consequently, it is important to develop efficient approximations to the multinomial NML. The topic of this section is to present three such methods. The first two of the methods, BIC and Rissanen's asymptotic expansion, are well-known, but the third one, called the Szpankowski approximation, is novel. Since we are able to compute the exact NML, it is also possible to assess the accuracy of the three approximations. This comparison is presented in Section 3.1.3.

In the following, we introduce the three approximations and instantiate them for the multinomial model family. It should be noted that BIC and Rissanen's asymptotic expansion are usually considered as approximations to the stochastic complexity, i.e., the negative logarithm of the NML. To make the formulas easier to interpret, we will adopt this established practice.

Bayesian Information Criterion: The *Bayesian Information Criterion (BIC)* [62], also known as the Schwarz criterion, is the simplest of the three approximations. As the name implies, the BIC has a Bayesian interpretation, but it can also be given a formulation in the MDL setting as showed in [55]. It is derived by expanding the log-likelihood function as a second order Taylor series around the maximum likelihood point $\hat{\boldsymbol{\theta}}$ and then integrating this expansion over the parameter space. This procedure is called the *Laplace's method*. The BIC formula is given by

$$-\log P_{\mathrm{BIC}}(\mathbf{x}^n \mid \mathcal{M}) = -\log P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n), \mathcal{M}) + \frac{d}{2}\log n + \mathcal{O}(1), \quad (3.3)$$

where d is the Euclidean dimension of the parameter space, i.e., the number of parameters. Looking at (3.3), we can see that it contains the same maximum likelihood term as the exact NML equation (2.3). Therefore, the second term $\frac{d}{2}\log(n)$ can be interpreted as an approximation to the parametric complexity.

The instantiation of the BIC approximation for the multinomial case is trivial. If the multinomial variable has K possible values, the number of parameters is $K-1$ and

$$-\log P_{\mathrm{BIC}}(\mathbf{x}^n \mid \mathcal{M}(K)) = -\log P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n), \mathcal{M}(K)) + \frac{K-1}{2}\log n + \mathcal{O}(1). \quad (3.4)$$

The main advantage of BIC is that it is very simple, intuitive and quick to compute. However, it is widely acknowledged that in model selection tasks BIC favors overly simple models (see, e.g., [68]).

Rissanen's Asymptotic Expansion: As proved in [56], for model classes that satisfy certain regularity conditions, a sharper asymptotic expansion than BIC can be derived for the NML. The most important regularity condition is that the Central Limit Theorem should hold for the maximum likelihood estimators for all the elements in the model class. The precise regularity conditions can be found in [56]. Rissanen's asymptotic expansion is given by

$$\begin{aligned} &-\log P_{\mathrm{RIS}}(\mathbf{x}^n \mid \mathcal{M}) = \\ &\quad -\log P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n), \mathcal{M}) + \frac{d}{2}\log\frac{n}{2\,\pi} + \log\int \sqrt{|I(\theta)|}d\theta + o\,(1), \end{aligned} \quad (3.5)$$

where the integral goes over the parameter space Θ. The matrix $I(\theta)$ is called the (expected) *Fisher information matrix* defined by

$$I(\theta) = -E_\theta\left[\frac{\partial^2 \log P(\mathbf{x}^n \mid \theta, \mathcal{M})}{\partial\theta_i\theta_j}\right], \quad (3.6)$$

where θ_i, θ_j go through all the possible pairs of parameters and the expectation is taken over the data space $\mathcal{X}$. The first two terms of (3.5) are essentially the same as in the BIC approximation (3.3). The crucial distinction is the integral term measuring the complexity that comes from the local geometrical properties of the model space. For a more precise discussion of the interpretation of this term, see [21]. Note that unlike the BIC approximation, Rissanen's expansion is *asymptotically correct*. This means that the error in the approximation vanishes as n goes to infinity.

Rissanen's asymptotic expansion for the $\mathcal{M}(K)$ model class was derived in [56], and it is given by

$$\begin{aligned} &-\log P_{\mathrm{RIS}}(\mathbf{x}^n \mid \mathcal{M}(K)) = \\ &-\log P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n), \mathcal{M}(K)) + \frac{K-1}{2}\log\frac{n}{2\pi} + \log\frac{\pi^{K/2}}{\Gamma(K/2)} + o\,(1), \end{aligned} \quad (3.7)$$

where $\Gamma(\cdot)$ is the *Euler gamma function* (see, e.g., [1]). This approximation is clearly very efficient to compute as well. Note that the derivation of the Rissanen's expansion for the Naive Bayes can be found in Paper I.

Szpankowski Approximation: An advanced mathematical tool called *singularity analysis* [16] can be used to derive an arbitrarily accurate ap-

proximation to the multinomial NML. Appendix A.4 gives a brief overview of the method. The Szpankowski approximation is based on a theorem on redundancy rate for memoryless sources [66], which gives

$$\begin{aligned} -\log P_{\mathrm{SZP}}(\mathbf{x}^n \mid \mathcal{M}(K)) = &-\log P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n), \mathcal{M}(K)) \\ &+ \frac{K-1}{2}\log\frac{n}{2} + \log\frac{\sqrt{\pi}}{\Gamma(K/2)} + \frac{\sqrt{2}K\cdot\Gamma(K/2)}{3\Gamma\left(\frac{K}{2}-\frac{1}{2}\right)}\cdot\frac{1}{\sqrt{n}} \\ &+ \left(\frac{3+K(K-2)(2K+1)}{36} - \frac{\Gamma^2(K/2)\cdot K^2}{9\Gamma^2\left(\frac{K}{2}-\frac{1}{2}\right)}\right)\cdot\frac{1}{n} \\ &+ \mathcal{O}\left(\frac{1}{n^{3/2}}\right). \end{aligned} \tag{3.8}$$

The full derivation of this approximation is given in Appendix B. Note that (3.8) is not a general NML approximation. It is only applicable for the multinomial case.

3.1.3 Comparison of the Approximations

As noted in the previous section, the ability to compute the exact NML for the multinomial model gives us a unique opportunity to test how accurate the NML approximations really are. The first thing to notice is that since all the three presented approximations contain the maximum likelihood term, we can ignore it in the comparisons and concentrate on the parametric complexity. Notice that we therefore avoid the problem of trying to choose representative and unbiased data sets for the experiments.

We conducted two sets of experiments with the three approximations. Firstly, we studied the accuracy of the approximations as a function of the size of the data n. In the second set of the experiments we varied the number of values of the multinomial variable. For all the experiments, the following names are used for the three approximations:

- BIC: Bayesian information criteria (3.4)
- RIS: Rissanen's asymptotic expansion (3.7)
- SZP: Szpankowski approximation (3.8)

The results of the first set of experiments can be seen in Figure 3.1, where the difference between the approximative and exact parametric complexity is plotted when the number of values K is set to 2, 4 and 9, respectively. In each figure the size of data n varies from 1 to 100. From the figures we can see that the SZP approximation is clearly the best of the

three. This is naturally anticipated since SZP is theoretically the most accurate one. What might be surprising is the absolute accuracy of SZP. The error is practically zero even for very small values of n. The second best of the approximations is RIS converging monotonically towards the exact value. However, this convergence gets slower when K increases. The figures also nicely show the typical behaviour of the BIC approximation. When the test setting becomes more complex (for $K > 3$), BIC starts to overestimate the parametric complexity.

In the second set of experiments we studied the accuracy of the three approximations when the number of values K varies from 2 to 10. Figure 3.2 shows the difference between the approximative and exact parametric complexity when the size of the data n is fixed to 25, 100 and 500, respectively. Naturally, the accuracy of the SZP approximation is superior in these tests as well. The most dramatic thing to notice from the figures is the rapid decrease in the accuracy of the BIC approximation when K increases. This is in contrast with the RIS approximation, which clearly gets more accurate with increasing amount of data, as anticipated by the theory.

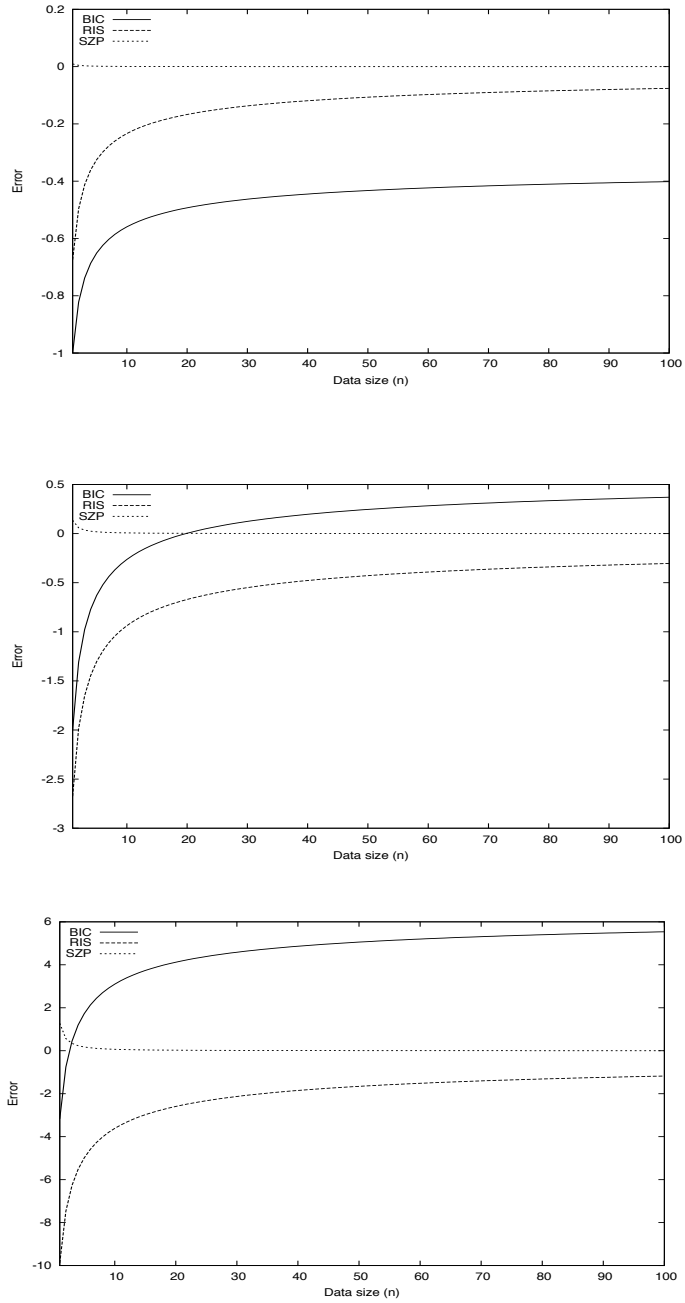

Figure 3.1: The accuracy of the three approximations as a function of the size of the data for $K = 2, 4$ and 9.

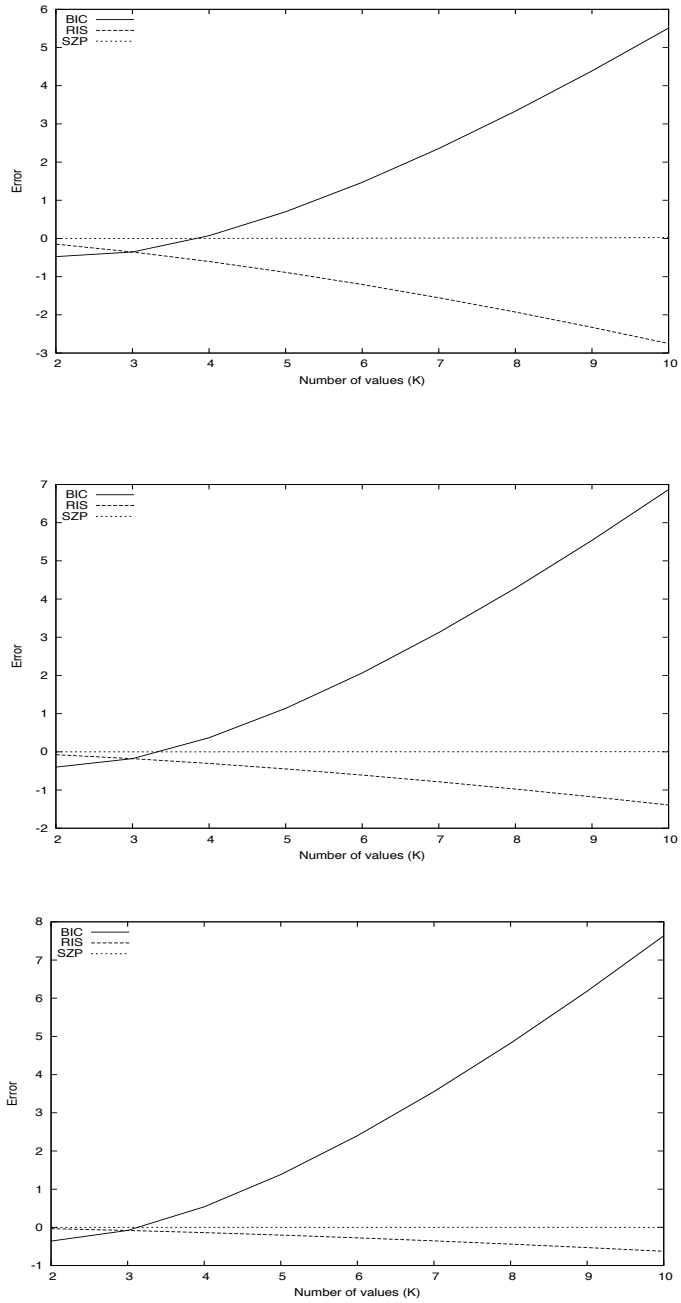

Figure 3.2: The accuracy of the three approximations as a function of the number of values. From top to bottom, the data size n is fixed to 25, 100 and 500.

3.2 The Naive Bayes Model Family

It is clear that the time complexity of computing the NML for the Naive Bayes model family (2.20) is also dominated by the parametric complexity $\mathcal{C}(\mathcal{M}(K_0, K_1, \ldots, K_m), n)$. It turns out (see Papers I and V) that the recursive formula (3.1) can be generalized to this case:

Theorem 3.2 *The terms $\mathcal{C}(\mathcal{M}(K_0, K_1, \ldots, K_m), n)$ satisfy the recurrence*

$$\mathcal{C}(\mathcal{M}(K_0, K_1, \ldots, K_m), n) = \sum_{r_1+r_2=n} \frac{n!}{r_1! r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \cdot \mathcal{C}_{NB}(\mathcal{M}(K^*, K_1, \ldots, K_m), r_1) \cdot \mathcal{C}_{NB}(\mathcal{M}(K_0 - K^*, K_1, \ldots, K_m), r_2), \tag{3.9}$$

where $K^* = 1, \ldots, K_0 - 1$.

Proof. See Papers I and V. □

In many practical applications of the Naive Bayes the quantity K_0 is unknown. Its value is typically determined as a part of the model class selection process. Consequently, it is necessary to compute NML for model classes $\mathcal{M}(K_0, K_1, \ldots, K_m)$, where K_0 has a range of values, say, $K_0 = 1, \ldots, K_{\max}$. The process of computing NML for this case is described in Algorithm 2. The time complexity of the algorithm is $\mathcal{O}\left(n^2 \cdot K_{\max}\right)$. If the value of K_0 is fixed, the time complexity drops to $\mathcal{O}\left(n^2 \cdot \log K_{\max}\right)$. See Paper V for more details.

Deriving accurate approximations to the Naive Bayes NML is more challenging than in the multinomial case. BIC and the Rissanen's asymptotic expansion can be computed for the Naive Bayes (see Paper I), but the equivalent of the Szpankowski approximation for the multinomial model family (3.8) has not been found. One simple approach is presented in Paper I, where the multinomial NML terms in Algorithm 2 are replaced by the approximations using (3.8). However, the time complexity of the resulting algorithm is still quadratic with respect to the size of the data.

Algorithm 2 The algorithm for computing the NML for the Naive Bayes model family for $K_0 = 1, \ldots, K_{\max}$.

Compute $\mathcal{C}(\mathcal{M}(k), j)$ for $k = 1, \ldots, V_{\max},\ j = 0, \ldots, n$, where $V_{\max} = \max\{K_1, \ldots, K_m\}$
for $K_0 = 1$ to $K_{\max}$ **do**
  Count the frequencies $h_1, \ldots, h_{K_0}, f_{ik1}, \ldots, f_{ikK_i}$ for $i = 1, \ldots, m,\ k = 1, \ldots, K_0$ from the data $\mathbf{x}^n$
  Compute the likelihood: $P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}(K_0, K_1, \ldots, K_m)))$ $= \prod_{k=1}^{K_0} \left(\frac{h_k}{n}\right)^{h_k} \prod_{i=1}^{m} \prod_{l=1}^{K_i} \left(\frac{f_{ikl}}{h_k}\right)^{f_{ikl}}$
  Set $\mathcal{C}(\mathcal{M}(K_0, K_1, \ldots, K_m), 0) = 1$
  if $K_0 = 1$ **then**
    Compute $\mathcal{C}(\mathcal{M}(1, K_1, \ldots, K_m), j) = \prod_{i=1}^{m} \mathcal{C}(\mathcal{M}(K_i), j)$ for $j = 1, \ldots, n$
  else
    Compute $\mathcal{C}(\mathcal{M}(K_0, K_1, \ldots, K_m), j)$ $= \sum_{r_1+r_2=j} \frac{j!}{r_1! r_2!} \left(\frac{r_1}{j}\right)^{r_1} \left(\frac{r_2}{j}\right)^{r_2} \cdot \mathcal{C}(\mathcal{M}(1, K_1, \ldots, K_m), r_1)$ $\cdot \mathcal{C}(\mathcal{M}(K_0 - 1, K_1, \ldots, K_m), r_2)$ for $j = 1, \ldots, n$
  end if
  Output $P_{\mathrm{NML}}(\mathbf{x}^n \mid \mathcal{M}(K_0, K_1, \ldots, K_m)) = \frac{P(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}(K_0, K_1, \ldots, K_m)))}{\mathcal{C}(\mathcal{M}(K_0, K_1, \ldots, K_m), n)}$
end for

Chapter 4

MDL Applications

In this chapter, we will show how the NML can be applied to practical problems using the techniques described in Chapter 3. Due to the computational efficiency problems, there are relatively few applications of NML. However, the existing applications have proven that NML works very well in practice and in many cases provides superior results when compared to alternative approaches.

We mention here some examples of NML applications. First, in Papers V and VI, NML was used for clustering of multi-dimensional data and its performance was compared to the Bayesian approach. The results showed that the performance of the NML was especially impressive with small sample sizes. Second, in [60], 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. Bioinformatical applications include [43] and [67], where NML was used for DNA sequence compression and data analysis in genomics, respectively. A scheme for using NML for histogram density estimation was presented in Paper IV. 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. Finally, in [12] NML histograms were used for modeling the attributes of the Naive Bayes classifier.

In the following, we will concentrate on two applications: histogram density estimation and clustering of multi-dimensional data. A computationally efficient NML approach for histogram density estimation was proposed in Paper IV. A theoretically interesting recursion formula derived in Paper III was shown to provide a way to compute the NML for histograms in linear time with respect to the sample size. The NML clustering framework was introduced in Paper V. The optimization aspect of the clustering prob-

lem was studied in Paper VI, where five algorithms for efficiently searching the exponentially-sized clustering space were empirically compared.

4.1 Histogram Density Estimation

Density estimation is one of the central problems in statistical inference and machine learning. Given a sample of observations, the goal of *histogram density estimation* is to find a piecewise constant density that describes the data best according to some pre-determined criterion. Although histograms are conceptually simple densities, they are very flexible and can model complex properties like multi-modality with a relatively small number of parameters. Furthermore, one does not need to assume any specific form for the underlying density function: given enough bins, a histogram estimator adapts to any kind of density.

The NML approach for irregular (variable-width bin) histogram density estimation described in Paper IV regards the problem as a model class selection task, where the possible sets of cut points (bin borders) are considered as model classes. The model parameters are the bin masses, or equivalently the bin heights. The NML criterion for comparing candidate histograms can be computed efficiently using the recursion formula derived in Paper III, where the problem of computing the parametric complexity for multinomial model was studied.

There is obviously an exponential number of different cut point sets. Therefore, a brute-force search is not feasible. In Paper IV it was shown that the NML-optimal cut point locations can be found via dynamic programming in a polynomial (quadratic) time with respect to the size of the set containing the cut points considered in the optimization process.

The histogram density estimation is naturally a well-studied problem, but unfortunately almost all of the previous studies, e.g. [6, 23, 73], consider regular (equal-width bin) histograms only. Most similar to our work is [59], in which irregular histograms are learned with the Bayesian mixture criterion using a uniform prior. The same criterion is also used in [23], but the histograms are equal-width only. It should be noted that this difference is significant as the Bayesian mixture criterion does not possess the optimality properties of the NML.

4.1.1 Definitions

Consider a sample of n outcomes $\mathbf{x}^n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ on the interval $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$. Without any loss of generality, we assume that the data is sorted into increasing order. Furthermore, we assume that the data is recorded at a

finite accuracy ϵ. This assumption is made to simplify the mathematical formulation, and as can be seen later, the effect of the accuracy parameter ϵ on the stochastic complexity is a constant that can be ignored in the model selection process.

Let $C = (c_1, \ldots, c_{K-1})$ be an increasing sequence of points partitioning the range $[\mathbf{x}_{\min} - \epsilon/2, \mathbf{x}_{\max} + \epsilon/2]$ into the following K intervals (bins):

$$([\mathbf{x}_{\min} - \epsilon/2, c_1],]c_1, c_2], \ldots,]c_{K-1}, \mathbf{x}_{\max} + \epsilon/2]). \tag{4.1}$$

The points c_k are called the *cut points* of the histogram. Define $c_0 = \mathbf{x}_{\min} - \epsilon/2, c_K = \mathbf{x}_{\max} + \epsilon/2$ and let $L_k = c_k - c_{k-1}, k = 1, \ldots, K$ be the bin lengths. Given a parameter vector $\theta \in \Theta$,

$$\Theta = \{(\theta_1, \ldots, \theta_K) : \theta_k \geq 0, \theta_1 + \cdots + \theta_K = 1\}, \tag{4.2}$$

and a set (sequence) of cut points C, we now define the histogram density f_h by

$$f_h(x \mid \theta, C) = \frac{\epsilon \cdot \theta_k}{L_k}, \tag{4.3}$$

where $x \in]c_{k-1}, c_k]$. Note that (4.3) does not define a density in the purest sense, since $f_h(x \mid \theta, C)$ is actually the probability that x falls into the interval $]x - \epsilon/2, x + \epsilon/2]$.

Given (4.3), the likelihood of the whole data sample $\mathbf{x}^n$ is easy to write. We have

$$f_h(\mathbf{x}^n \mid \theta, C) = \prod_{k=1}^{K} \left(\frac{\epsilon \cdot \theta_k}{L_k} \right)^{h_k}, \tag{4.4}$$

where h_k is the number of data points falling into bin k.

4.1.2 NML Histogram

To instantiate the NML distribution (2.3) for the histogram density f_h, we need to find the maximum likelihood parameters $\hat{\boldsymbol{\theta}}(x^n) = (\hat{\theta}_1, \ldots, \hat{\theta}_K)$ and an efficient way to compute the parametric complexity. It is well-known that the ML parameters are given by the relative frequencies $\hat{\theta}_k = h_k/n$, so that we have

$$f_h(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n), C) = \prod_{k=1}^{K} \left(\frac{\epsilon \cdot h_k}{L_k \cdot n} \right)^{h_k}. \tag{4.5}$$

Denote now the parametric complexity of a K-bin histogram by $\log \mathcal{C}(H_K, n)$. We now have the following theorem:

Theorem 4.1 *The term $\mathcal{C}(H_K, n)$ is given by*

$$\mathcal{C}(H_K, n) = \sum_{h_1+\cdots+h_K=n} \frac{n!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k}, \tag{4.6}$$

i.e., the same as the parametric complexity of a K-valued multinomial model.

Proof. See research paper IV. □

This result means that we can compute the parametric complexity for histogram densities using Algorithm 1.

We are now ready to write down the stochastic complexity (2.6) for the histogram model. We have

$$\mathrm{SC}(\mathbf{x}^n \mid C) = -\log \frac{\prod_{k=1}^{K} \left(\frac{\epsilon \cdot h_k}{L_k \cdot n}\right)^{h_k}}{\mathcal{C}(H_K, n)} \tag{4.7}$$

$$= \sum_{k=1}^{K} -h_k(\log(\epsilon \cdot h_k) - \log(L_k \cdot n)) + \log \mathcal{C}(H_K, n). \tag{4.8}$$

Equation (4.8) is the basis for measuring the quality of NML histograms, i.e., comparing different cut point sets. It should be noted that as the term $\sum_{k=1}^{K} -h_k \log \epsilon = -n \log \epsilon$ is a constant with respect to C, the value of ϵ does not affect the comparison.

The histogram density estimation problem is now straightforward to define: find the cut point set C which optimizes the given goodness criterion. In our case the criterion is based on the stochastic complexity (4.8), and the cut point sets are considered as model classes. In practical model class selection tasks, however, the stochastic complexity criterion itself may not be sufficient. The reason is that it is also necessary to encode the model class index in some way, as argued in [21]. We assume that the model class index is encoded with a uniform distribution over all the cut point sets of the same size. For a K-bin histogram with E possible cut points, there are clearly $\binom{E}{K-1}$ ways to choose the cut points. Thus, the codelength for encoding them is $\log \binom{E}{K-1}$.

After these considerations, we define the final criterion (or score) used for comparing different cut point sets as

$$B(\mathbf{x}^n \mid E, K, C) = \mathrm{SC}(\mathbf{x}^n \mid C) + \log \binom{E}{K-1}$$

$$= \sum_{k=1}^{K} -h_k \left(\log(\epsilon \cdot h_k) - \log(L_k \cdot n)\right) + \log \mathcal{C}(H_K, n) + \log \binom{E}{K-1}. \tag{4.9}$$

It is clear that there are an exponential number of possible cut point sets, and thus an exhaustive search to minimize (4.9) is not feasible. However, the optimal cut point set can be found via dynamic programming, which works by tabulating partial solutions to the problem. The final solution is then found recursively. For details, see Paper IV.

To demonstrate the behaviour of the NML histogram method in practice we implemented the dynamic programming algorithm and ran some simulation tests (see Paper IV). We generated data samples of various size from densities of different shapes and then used the dynamic programming method to find the NML-optimal histograms. Figure 4.1 shows two examples of the generating densities (labeled gm6 and gm8) and the corresponding NML-optimal histograms. The sample size is fixed to 10000, and

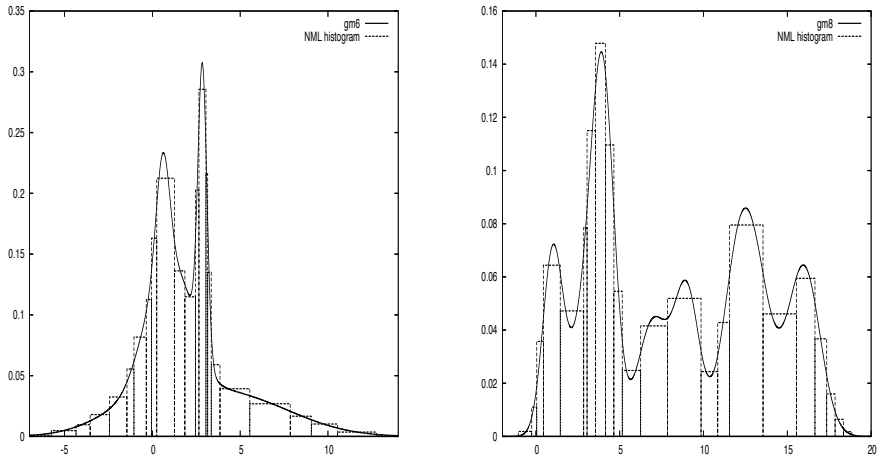

Figure 4.1: The generating densities gm6 and gm8 and the corresponding NML-optimal histograms.

the generating densities are Gaussian finite mixtures with 6 and 8 components, respectively. From the plots we can see that the NML histogram method is able to capture properties such as multi-modality and long tails. Another nice feature is that the algorithm automatically places more bins to the areas where more detail is needed like the high, narrow peaks of gm6. See Paper IV for more empirical tests and discussion.

4.2 Clustering

A *clustering* is a partitional data assignment or data labeling problem, where the goal is to partition the data into mutually exclusive clusters so that similar data vectors are grouped together. The number of clusters is unknown, and determining the optimal number is part of the clustering problem. The data are assumed to be in a vector form so that each data item is a vector consisting of a fixed number of attribute values. Within this framework two fundamental problems can be identified: how to define the goodness of a clustering (data partitioning) and how to find good clusterings with respect to the chosen scoring criterion.

Traditionally, the scoring problem has been approached by first fixing a distance metric, and then by defining a global goodness measure based on this distance metric. However, although this approach is intuitively quite appealing, from the theoretical point of view it introduces many problems such as choosing a suitable distance metric and the handling of non-continuous attributes. A completely different approach to clustering is offered by the *model-based approach*, where for each cluster a data generating function (a probability distribution) is assumed, and the clustering problem is defined as the task to identify these distributions (see, e.g., [64, 18, 7]). In other words, the data are assumed to be generated by a finite mixture model [13, 69, 44]. In this framework the optimality of a clustering can be defined as a function of the fit of data with the finite mixture model, not as a function of the distances between the data vectors.

In Paper V we proposed an NML-based approach for clustering. Intuitively, the idea is that a good clustering is such that one can encode the cluster labels *together* with the data so that the resulting code length is minimized. When the cluster labels are fixed, the finite mixture model is essentially the same as the Naive Bayes model, which allows the use of the techniques described in Section 3.2 for efficient computation of the NML criteria.

The optimization part of the clustering problem, i.e., how to find good clusterings with respect to the NML score, was studied in Paper VI. In that work, five algorithms were proposed to the problem and their performance was compared via empirical tests using several real-world datasets. In Section 4.2.2 we shortly summarize these empirical results.

4.2.1 NML Clustering

Let us assume that our problem domain consists of m discrete variables $X_1, \ldots, X_m$ and that the variable X_i has K_i values. The data $\mathbf{x}^n =$

$(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ consists of observations $\mathbf{x}_j = (x_{j1}, \ldots, x_{jm}) \in \mathcal{X}$, where

$$\mathcal{X} = \{1, 2, \ldots, K_1\} \times \cdots \times \{1, 2, \ldots, K_m\}. \tag{4.10}$$

We assume that the possibly originally continuous variables have been discretized. One reason for focusing on discrete data is that in this case we can model the domain variables by multinomial distributions without having to make restricting assumptions about unimodality, normality etc., which is the situation we face in the continuous case.

A *clustering* of the data set $\mathbf{x}^n$ is defined as a partitioning of the data into mutually exclusive subsets, the union of which forms the data set. The number of subsets is a priori unknown. The *clustering problem* is the task to determine the number of subsets, and to decide to which cluster each data vector belongs.

Formally, we can notate a clustering by using a *clustering vector* $z^n = (z_1, \ldots, z_n)$, where z_j denotes the index of the cluster to which the data vector $\mathbf{x}_j$ is assigned to. Denote the *clustering variable* by X_0 so that z^n is a sample from the distribution of X_0. The number of clusters, say K_0, is implicitly defined in the clustering vector, as it can be determined by counting the number of different values appearing in z^n.

In Paper V, we suggested the following NML-based criterion for finding the optimal clustering $\hat{z}^n$:

$$\hat{z}^n = \arg\max_{z^n} P_{\mathrm{NML}}(\mathbf{x}^n, z^n \mid \mathcal{M}(K_0, K_1, \ldots, K_m)), \tag{4.11}$$

where $\mathcal{M}(K_0, K_1, \ldots, K_m)$ is the Naive Bayes model family with K_0 components. In the clustering framework this means that the data vectors should be partitioned so that the vectors belonging to the same cluster can be compressed well together, i.e., that those data vectors that obey the same set of underlying regularities are grouped together.

Naturally, the criteria for comparing different clusterings can be based on other approaches like Bayesian statistics. In the Bayesian case, the NML distribution in (4.11) is replaced by the Bayesian marginal likelihood (see, e.g. [8, 24]). The approaches were compared empirically in Paper V, where it was shown that NML produces the best results especially with small sample sizes.

4.2.2 Comparison of Clustering Algorithms

The space of potential clusterings is obviously exponential in size, which means that in practice we need to resort to combinatorial search algorithms in our attempt to solve the clustering problem. The search algorithm used

in the empirical tests in Paper V was a simple stochastic greedy algorithm. In Paper VI, we compared five different algorithms for finding good clusterings using several real-world datasets from the UCI repository [2]. Two sets of results were presented. The first set concentrates on finding the number of clusters and the actual clustering minimizing the NML score (4.11). In the second set of experiments, we tested how long it takes for each of the five algorithms to find the respective maximum NML value.

The first search algorithm candidate is a simple stochastic greedy (SG) algorithm suggested in Paper V. Since our definition of clustering is based on the finite mixture model, the standard mixture learning algorithm, EM (Expectation-Maximization) (See [11, 41]) is a natural choice as a second clustering algorithm. The third candidate algorithm is the K-means algorithm (KM), sometimes called the CEM algorithm [45], is a simple modification to the EM algorithm.

Each of the algorithms mentioned above needs to be initialized prior to the iterative updating procedure. In our tests, we started each algorithm simply by choosing a random clustering. To test the importance of the initialization, we added two hybrid methods to our set of candidate search algorithms. The first hybrid algorithm (KMSG) starts by running the K-means algorithm until convergence and then switches to the stochastic greedy search. The second algorithm (EMSG) is the same except that the EM algorithm is used as an initializer.

Having fixed the set of candidate search algorithms, the next task is to define a strategy for finding the optimal number of clusters and the actual clustering. Since all the five algorithms converge to a local optimum of the stochastic complexity, the natural strategy is to restart the algorithms several times from different starting points.

Although the NML scoring criterion can be used for comparing clusterings with different number of clusters, the framework does not offer an explicit way to directly infer the optimal number of clusters (K). Consequently, the second part of our search strategy is to vary the parameter K. The complete search strategy is described in Algorithm 3.

In the first batch of results we tested which of the five algorithms find the best clusterings in terms of the stochastic complexity. The results showed that all five candidate algorithms end up choosing a similar number of clusters. However, when we looked at the actual SC values, there were significant differences between the algorithms. Since SC can be interpreted as a quality of a clustering, these differences are important. The hybrid EMSG was clearly the best one of the algorithms, especially with more complex cases, i.e., when the size of data and the optimal number of clusters

Algorithm 3 The search strategy used in our tests.

```
repeat
  for all D in datasets do
    for K = 1 to 20 do
      Choose a random initial K-clustering for dataset D
      for all A in {SG, KM, EM, KMSG, EMSG} do
        Run the algorithm A until converged
      end for
    end for
  end for
until 50 restarts have been made
```

was bigger. Another interesting observation is that the traditional KM and EM algorithms were clearly the worst of the candidate algorithms.

In the second set of experiments we recorded how much CPU time each algorithm required for finding their respective optimal clustering. The most important thing to notice from these results was that the hybrid EMSG algorithm, which in the first batch of empirical results was found to produce comparable or better results than SG, was almost always significantly faster than the SG algorithm proving the intuitive argument that choosing a good initial clustering is important. This made the EMSG algorithm a clear overall winner in our experiments. It is also noteworthy that KM and EM were often much slower than the other algorithms even though they produced inferior results. This makes the applicability of KM and EM even more questionable in the setting used here. See Paper VI for all the details of the empirical tests.

Chapter 5

Conclusion

The Normalized Maximum Likelihood (NML) distribution offers a universal, minimax optimal approach to statistical modeling. In this thesis we have surveyed efficient algorithms for computing the NML in the case of discrete data sets and two model families of practical importance. The first model family we discussed is the multinomial, which can be applied to problems such as density estimation and discretization. In this case, the NML can be computed in linear time. For the Naive Bayes model family, the NML can be computed in quadratic time. Models of this type have been used extensively in clustering or classification domains with good results.

To demonstrate the applicability of the computation algorithms presented, we also discussed two NML applications. The first application was an information-theoretic framework for histogram density estimation. The selected approach based on the MDL principle has several advantages. Firstly, the MDL criterion for model class selection (stochastic complexity) has nice theoretical optimality properties. Secondly, by regarding histogram density estimation as a model class selection problem, it is possible to learn generic, variable-width bin histograms and also estimate the optimal bin count automatically. Furthermore, the MDL criterion itself can be used as a measure of quality of a density estimator, which means that there is no need to assume anything about the underlying generating density. Since the model selection criterion is based on the NML distribution, there is also no need to specify any prior distribution for the parameters.

The second application we described was NML clustering of data. We suggested a framework for this problem based on the idea that a good clustering is such that it allows efficient compression when the data are encoded together with the cluster labels. We also introduced five optimization algorithms for minimizing the stochastic complexity. Using these algorithms, we

conducted an extensive set of experiments with several real-world datasets. In the first part of the tests we recorded the number of clusters chosen and the quality of the actual clusterings found by the algorithms, while the idea of the second batch of tests was to see how much CPU time each algorithm requires for finding the best solution. In the empirical results we found out that all the five algorithms were useful if the goal is to find the NML-optimal number of clusters. However, the quality of the individual clusterings found by the more traditional KM and EM algorithms was questionable. These algorithms were also found to be slow. The most interesting observation was that the novel hybrid EMSG algorithm produced the best results and was also fast.

The methods presented are especially suitable for problems that involve multi-dimensional discrete data sets. Furthermore, unlike the Bayesian methods, information-theoretic approaches such as ours do not require a prior for the model parameters. This is a most important aspect, as constructing a reasonable parameter prior is a notoriously difficult problem, particularly in domains with little background knowledge. All in all, information theory has been found to offer a natural and successful theoretical framework for applications in general.

In the future, our plan is to extend the current work to more complex cases such as general Bayesian networks, which would allow the use of NML in even more involved modeling tasks. Another natural area of future work is to apply the methods of this thesis to other practical tasks involving large discrete databases and compare the results to other approaches, such as those based on Bayesian statistics.

Appendices

Chapter A

Mathematical Background

The purpose of this appendix is to provide the reader with some mathematical techniques that are used in the other parts of the thesis, especially in Appendix B. The topics covered are complex analysis, formal power series, generating functions and asymptotic analysis.

A.1 Review of Complex Analysis

The theory of functions of a complex variable, also called complex analysis for brevity, is one of the most beautiful as well as useful branches of mathematics. It is an essential part of the mathematical background of physicists, mathematicians, engineers and other scientists. From the theoretical viewpoint this is because many mathematical concepts become clarified and unified when examined in the light of complex analysis. From the applied viewpoint the theory is of tremendous value in the solution of problems such as fluid dynamics, heat flow, aerodynamics, electromagnetic theory and many other fields of science and engineering.

For a computer scientist, the importance of complex analysis comes from the fact that the theory can be applied to, e.g., calculation of finite and infinite sums, analyzing algorithms and finding asymptotic behaviour of sequences. In this thesis complex analysis is used for deriving the accurate NML approximation in Appendix B. The purpose of this appendix is to briefly review the most relevant definitions and theorems of complex analysis. For further reading on the subject we recommend the books [51, 74, 65, 26].

A.1.1 The Complex Numbers and the Complex Plane

The set $\mathbb{C}$ of *complex numbers* is introduced to permit solutions to equations like

$$x^2 + 1 = 0, \tag{A.1}$$

that has no solutions in the set $\mathbb{R}$ of real numbers. A complex number has the form $a+bi$, where a and b are real numbers and i is called the imaginary unit and has the property $i^2 = -1$. If $z = a + bi$, a is called the *real part* of z and b is called the *imaginary part* of z. The symbol z, which can stand for any of a set of complex numbers, is called a *complex variable*.

A complex number $z = a+bi$ is uniquely determined by an ordered pair of real numbers (a, b). Because of this correspondence we can associate z with a point (a, b) in coordinate plane. This plane is then called the *complex plane*. The horizontal or x-axis is called the *real axis* and the vertical or y-axis is called the *imaginary axis*. If P is a point in the complex plane corresponding to the complex number $z = a + bi$, then we see from Figure A.1 that

$$a = r\cos\theta, \quad b = r\sin\theta, \tag{A.2}$$

where $r = \sqrt{a^2 + b^2} = |a + bi|$ is called the *modulus* or *absolute value* of z, and θ is called the *argument* of z. It follows that we can write

$$z = a + bi = r(\cos\theta + i\sin\theta), \tag{A.3}$$

which is called the *polar form* of the complex number z.

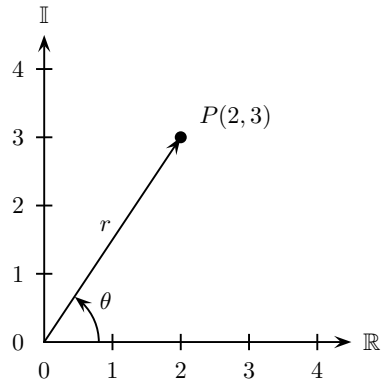

Figure A.1: The polar form of complex number $2 + 3i$.

A.1.2 Roots of Complex Numbers

A number w is called an nth root of a complex number z if $w^n = z$, and we write $w = z^{1/n}$. We can show that if n is a positive integer, then

$$z^{1/n} = (r(\cos\theta + i\sin\theta))^{1/n} \tag{A.4}$$

$$= r^{1/n}\left[\cos\left(\frac{\theta + 2k\pi}{n}\right) + i\sin\left(\frac{\theta + 2k\pi}{n}\right)\right], \tag{A.5}$$

for $k = 0, 1, 2, \ldots, n-1$. It follows that there are n different values for $z^{1/n}$. For example, the five 5th roots of number 32 are

- 2
- $2\left(\cos\frac{2\pi}{5} + i\sin\frac{2\pi}{5}\right)$
- $2\left(\cos\frac{4\pi}{5} + i\sin\frac{4\pi}{5}\right)$
- $2\left(\cos\frac{6\pi}{5} + i\sin\frac{6\pi}{5}\right)$
- $2\left(\cos\frac{8\pi}{5} + i\sin\frac{8\pi}{5}\right)$,

as illustrated in Figure A.2.

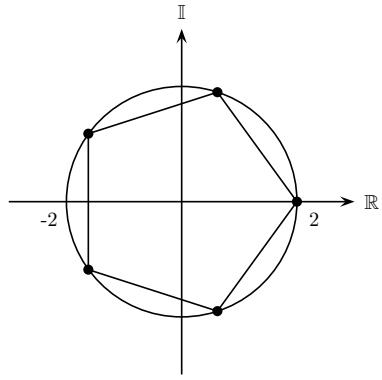

Figure A.2: The 5th roots of complex number 32.

Note that the roots lie on a circle centered at origin of radius $r = 2$ and are spaced at equal angular intervals of $2\pi/5$ radians, i.e., they represent the vertices of a regular pentagon.

A.1.3 Analytic Functions

A *complex function* is a function f whose domain and range are subsets of the set $\mathbb{C}$ of complex numbers. Because $\mathbb{R}$ is a subset of the set $\mathbb{C}$, every real-valued function of a real variable is also a complex function. Furthermore, every complex function can be defined in terms of two real functions $u(a, b)$ and $v(a, b)$ as $f(z) = u(a, b)+iv(a, b)$. This implies that the study of complex functions is closely related to the study of real multivariate functions of two real variables.

Suppose that a complex function f is defined in a deleted neighborhood of a point z_0 and that l is a complex number. The *limit* of f as z tends to z_0 exists and is equal to l, written as $\lim_{z\to z_0} f(z) = l$, if for every $\epsilon > 0$ there exists a number δ such that $|f(z) - l| < \epsilon$ whenever $|z - z_0| < \delta$. Complex and real limits have many common properties, but there is at least one very important difference. For limits of complex functions, z is allowed to approach z_0 from any direction in the complex plane, that is, along any curve or path through z_0. In order that $\lim_{z\to z_0} f(z) = l$, it is required that $f(z)$ approaches the same complex number l along every possible curve through z_0.

The complex derivative is defined similarly as its real counterpart. Suppose that a complex function f is defined in a neighborhood of a point z_0. The *derivative* of f at z_0 is

$$f'(z_0) = \lim_{z\to z_0} \frac{f(z) - f(z_0)}{z - z_0}, \tag{A.6}$$

provided that the limit exists. Furthermore, the function f is said to be *analytic* at a point z_0 if the derivative $f'(z_0)$ exists at z_0 and at every point in some neighborhood of z_0. If f is analytic at every point in an open connected set (domain) D we say that $f(z)$ is analytic in D. The term *holomorphic* is often used as a synonym for analytic. A function that is analytic at every point in the complex plane is said to be an *entire function*.

A remarkable property of analytic functions is the *infinite differentiability*: if f is analytic in a domain D, then f has derivatives of *all* orders in D. This is not necessarily true for functions of real variables. Furthermore, if z_0 is a point in D, then by the *Taylor's theorem*, f has the series representation

$$f(z) = \sum_{n\geq 0} \frac{f^{(n)}(z_0)}{n!}(z - z_0)^n \tag{A.7}$$

valid for the largest circle C with center at z_0 and radius R that lies entirely within D. The number R is called the *radius of convergence*.

A.1.4 Complex Integration

A complex integral is defined in a manner that is quite similar to that of a line integral in the Cartesian plane. Let f be a complex function defined at all points on a smooth curve C. Subdivide C into n parts by means of $z_1, \ldots, z_{n-1}$ chosen arbitrarily. On each arc joining z_{k-1} and z_k choose a point α_k and form a sum

$$S_n = f(\alpha_1)(z_1 - z_0) + f(\alpha_2)(z_2 - z_1) + \cdots + f(\alpha_n)(z_n - z_{n-1}), \quad \text{(A.8)}$$

where z_0 and z_n are the starting and end poinds of C, respectively. On writing $\Delta z_k = z_k - z_{k-1}$, this becomes

$$S_n = \sum_{k=1}^{n} f(\alpha_k)\Delta z_k. \quad \text{(A.9)}$$

Let the number of subdivisions n increase in such a way that the largest of the arc lengths $|\Delta z_k|$ approaches zero. If the sum S_n approaches a limit which does not depend on the choice of the z_k's we call this limit a *complex (line) integral* of f along curve C and denote it by

$$\oint_C f(z)dz. \quad \text{(A.10)}$$

Function f is said to be *integrable* along curve C. If f is analytic at all points of a domain D and if curve C is lying in D then f is certainly integrable along C.

Another remarkable result of complex analysis is the *Cauchy's integral theorem*: Suppose that a function f is analytic at all points within and on a simple closed curve C. Then,

$$\oint_C f(z)dz = 0. \quad \text{(A.11)}$$

A.1.5 Laurent Expansion

If a complex function f fails to be analytic at a point z_0, then this point is said to be a *singularity* of the function f. The Taylor expansion (A.7) does not hold at a singularity point. However, if the singularity z_0 is *isolated*, i.e., there exists some deleted neighborhood of z_0 throughout which f is analytic, it is possible to represent f by a series involving both negative and non-negative integer powers of $z - z_0$. This series is called the *Laurent expansion*,

$$f(z) = \sum_{n=-\infty}^{\infty} a_n(z - z_0)^n. \quad \text{(A.12)}$$

Furthermore, the coefficients a_n are given by

$$a_n = \frac{1}{2\pi i} \oint_C \frac{f(z)dz}{(z - z_0)^{n+1}}, \tag{A.13}$$

where C is any simple closed curve that encloses z_0 and that lies entirely inside a region in which f is analytic.

An isolated singularity z_0 of a complex function f is given a classification depending on whether its Laurent expansion (A.12) contains zero, a finite number, or an infinite number of terms of negative powers.

1. If all the coefficients a_{-n} are zero, then z_0 is called a *removable singularity.*

2. If a finite number, say k, of coefficients a_{-n} are non-zero, then z_0 is called a *pole of order* k.

3. If an infinite number of coefficients a_{-n} are non-zero, then z_0 is called an *essential singularity.*

If the denominator of a rational function f has a zero of order k at z_0, then the function f has a pole of order k at z_0.

A.1.6 The Residue Theorem

The coefficient a_{-1} in the Laurent series (A.12) has a special meaning. This coefficient is called the *residue* of function f at the isolated singularity z_0 and denoted by

$$a_{-1} = \operatorname*{Res}_{z=z_0} f(z). \tag{A.14}$$

The reason why the residue concept is important is that under some circumstances we can evaluate complex integrals by summing the residues at the isolated singularities of a function. More precisely, the *Residue theorem* states that if f is analytic inside and on a simple closed curve C, except at a finite number of isolated singularities $z_1, z_2, \ldots, z_n$ within C, then

$$\oint_C f(z)dz = 2\pi i \sum_{k=1}^{n} \operatorname*{Res}_{z=z_k} f(z). \tag{A.15}$$

Note that the residue theorem is an extension of the Cauchy's integral theorem (A.11).

The residue theory has many applications. It can be used, e.g., to evaluate *real* integrals, to find the locations of zeros of an analytic function,

to sum infinite series and to find integral transforms such as the Laplace transform and its inverse.

There are several ways to calculate residues. Obviously, if we can somehow find the Laurent expansion of a function f at point z_0, we can just pick the coefficient a_{-1} from the series. Otherwise, if the singularity z_0 is a pole of order k, then

$$\operatorname*{Res}_{z=z_0} f(z) = \frac{1}{(k-1)!} \lim_{z \to z_0} \frac{d^{k-1}}{dz^{k-1}} [(z - z_0)^k f(z)]. \tag{A.16}$$

Interestingly, this means that in some cases complex integrals can be evaluated by taking derivatives of complex functions.

A.1.7 Puiseux Expansion

We finalize the discussion on complex analysis by a very special topic of fractional power or *Puiseux* series. This series is relevant in the derivation of the accurate NML approximation in Appendix B. Suppose f is a multivalued analytic function and z_0 its special singularity called *branch point* of order $k-1$. The exact definition of a branch point is complicated and omitted here, but as an example the function $(z-1)^{1/3}$ has a branch point of order 2 at $z_0 = 1$, and the function $\sqrt{z(z-1)}$ has two branch points at 0 and 1, each of order 1. In the neighborhood of a branch point z_0, the function f can be represented as a series

$$f(z) = \sum_{n=-\infty}^{\infty} a_n (z - z_0)^{n/k}. \tag{A.17}$$

Note that the series (A.17) is an extension of the Laurent expansion (A.12).

Unfortunately, there is no simple formula for calculating the coefficients of a Puiseux series. For the purposes of this thesis, however, a special result on inversion of Puiseux series presented in [14] is suitable. In that work, series expansions are classified into four types of systematic patterns. We omit the full categorization here, but the category relevant to the main part of the thesis is called "Type II" and it is of form

$$f(z) = a_0 + \sum_{n \geq 1} a_n (z - z_0)^{n-1+\beta}, \tag{A.18}$$

where $\beta > 0$. According to the theorem, the inverse function of f can then be represented as a Puiseux series

$$F(w) = \sum_{n \geq 0} b_n (w - w_0)^{n/\beta}, \tag{A.19}$$

for some sequence of coefficients b_n. Note that $f(z_0) = a_0 = w_0$ and $F(w_0) = z_0$. An example of using the inversion is given in Appendix B.

A.2 Formal Power Series

In this appendix we give a short overview of the theory of formal power series. We concentrate on the issues that are relevant to the other parts of the thesis. Readers interested to learn more about formal power series can refer to, e.g., [71, 19].

A.2.1 Definition

A *formal power series* is an expression of the form

$$\sum_{n\geq 0} a_n z^n, \tag{A.20}$$

where the numbers a_n are called the coefficients of the series. In the theory of formal power series, the variable z is considered as a formal symbol, and the convergence of series (A.20) is not an issue. If, however, the series converges for some values of z, it is a big advantage. For example, the singularity analysis discussed in Appendix A.4 is based on this *analytic theory* of power series. In practice, however, all the operations on series can be performed without worrying about the convergence.

A.2.2 Linear Combination

The most basic of formal power series operations is taking a linear combination of two series. Since formal power series are just infinite polynomials, we have

$$\alpha \sum_{n\geq 0} a_n z^n + \beta \sum_{n\geq 0} b_n z^n = \sum_{n\geq 0} (\alpha a_n + \beta b_n)\, z^n, \tag{A.21}$$

for numbers α, β.

A.2.3 Multiplication

Another basic operation is multiplication of two or more power series. By basic arithmetics,

$$\left(\sum_{n\geq 0} a_n z^n\right) \cdot \left(\sum_{n\geq 0} b_n z^n\right) = (a_0 + a_1 z + a_2 z^2 + \cdots)(b_0 + b_1 z + b_2 z^2 + \cdots) \tag{A.22}$$

$$= (a_0 b_0) + (a_0 b_1 + a_1 b_0) z \tag{A.23}$$
$$+ (a_0 b_2 + a_1 b_1 + a_2 b_0) z^2 + \cdots$$

$$= \sum_{n\geq 0} \left(\sum_{k=0}^{n} a_k b_{n-k}\right) z^n. \tag{A.24}$$

The series (A.24) is called the *Cauchy product* or *convolution*.

The multiplication operation also generalizes to a product of three or more series. For example, the product of three formal power series is

$$\left(\sum_{n\geq 0} a_n z^n\right) \cdot \left(\sum_{n\geq 0} b_n z^n\right) \cdot \left(\sum_{n\geq 0} c_n z^n\right) \tag{A.25}$$

$$= (a_0 + a_1 z + a_2 z^2 + \cdots)(b_0 + b_1 z + b_2 z^2 + \cdots)(c_0 + c_1 z + c_2 z^2 + \cdots) \tag{A.26}$$

$$= (a_0 b_0 c_0) + (a_0 b_0 c_1 + a_0 b_1 c_0 + a_1 b_0 c_0) z \tag{A.27}$$
$$+ (a_0 b_0 c_2 + a_0 b_1 c_1 + a_0 b_2 c_0 + a_1 b_0 c_1 + a_1 b_1 c_0 + a_2 b_0 c_0) z^2 + \cdots \tag{A.28}$$

$$= \sum_{n\geq 0} \left(\sum_{r+s+t=n} a_r b_s c_t\right) z^n. \tag{A.29}$$

A.2.4 Reciprocal Series

A more complex operation is taking the reciprocal of a formal power series. It is defined as

$$\sum_{n\geq 0} b_n z^n = \frac{1}{\sum_{n\geq 0} a_n z^n}, \tag{A.30}$$

from which it follows that

$$(a_0 + a_1 z + a_2 z^2 + \cdots)(b_0 + b_1 z + b_2 z^2 + \cdots) \equiv 1, \tag{A.31}$$

i.e., the trivial sequence (1, 0, 0, ...). Using the product rule (A.24) we can solve the reciprocal coefficients b_n as

$$a_0 b_0 = 1, \quad b_0 = \frac{1}{a_0} \tag{A.32}$$

$$a_0 b_1 + a_1 b_0 = 0, \quad b_1 = -\frac{a_1 b_0}{a_0} = -\frac{a_1}{a_0^2} \tag{A.33}$$

$$a_0 b_2 + a_1 b_1 + a_2 b_0 = 0, \quad b_2 = -\frac{a_1 b_1 + a_2 b_0}{a_0} = \frac{a_1^2}{a_0^3} - \frac{a_2}{a_0^2}, \tag{A.34}$$

and so on. This result is used in Appendix B. It is easy to see that the reciprocal of a series is only defined when a_0, the constant term in the original series, is non-zero.

As a simple example, we show that the reciprocal of $(1, -1, 0, 0, \ldots)$ is the sequence $(1, 1, 1, \ldots)$, i.e.,

$$\frac{1}{1-z} = \sum_{n \geq 0} z^n. \tag{A.35}$$

This is easy to prove, since

$$(1-z)(1+z+z^2+\cdots) = (1+z+z^2+\cdots) + (-z-z^2+\cdots) \equiv 1. \tag{A.36}$$

A.2.5 Inverse Series

The reciprocal operation is not to be confused with the subtler operation of inverting a series. Inverse of a series

$$f(z) = \sum_{n \geq 0} a_n z^n \tag{A.37}$$

is defined as a series

$$g(z) = \sum_{n \geq 0} b_n z^n, \tag{A.38}$$

if

$$f(g(z)) = g(f(z)) \tag{A.39}$$

$$\begin{aligned} &= a_0 + a_1(b_0 + b_1 z + b_2 z^2 + \cdots) \\ &\qquad + a_2(b_0 + b_1 z + b_2 z^2 + \cdots)^2 + \cdots \equiv z, \end{aligned} \tag{A.40}$$

i.e., the trivial sequence (0, 1, 0, 0, ...). As argued in [71] (Chapter 2.1), this operation only makes sense if the constant terms a_0, b_0 are zero or if f

is a polynomial (finite). Otherwise, the process of finding the coefficients of the inverse series is infinite. Consequently, we have

$$f(g(z)) = a_1(b_1z + b_2z^2 + b_3z^3 + \cdots) + a_2(b_1z + b_2z^2 + b_3z^3 + \cdots)^2 \tag{A.41}$$

$$+ a_3(b_1z + b_2z^2 + b_3z^3 + \cdots)^3 + \cdots$$

$$= (a_1b_1)z + (a_1b_2 + a_2b_1^2)z^2 + (a_1b_3 + 2a_2b_1b_2 + a_3b_1^3)z^3 + \cdots \equiv z, \tag{A.42}$$

from which we get by coefficient comparison

$$a_1b_1 = 1, \quad b_1 = \frac{1}{a_1} \tag{A.43}$$

$$a_1b_2 + a_2b_1^2 = 0, \quad b_2 = -\frac{a_2b_1^2}{a_1} = -\frac{a_2}{a_1^3} \tag{A.44}$$

$$a_1b_3 + 2a_2b_1b_2 + a_3b_1^3 = 0, \quad b_3 = -\frac{2a_2b_1b_2 + a_3b_1^3}{a_1} = \frac{2a_2^2}{a_1^5} - \frac{a_3}{a_1^4}. \tag{A.45}$$

This result is also used in Appendix B.

A.3 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 finding recurrence formulas and asymptotic expansions, proving combinatorial identities and finding statistical properties of a sequence.

In this appendix we will present a short overview of generating functions and illustrate their use with several examples. Good sources for further reading on generating functions are [71, 3, 19, 27, 28, 29].

A.3.1 Definition

The (ordinary) generating function of a sequence

$$\langle a_n \rangle = (a_0, a_1, a_2, \ldots) \tag{A.46}$$

is defined as a series

$$A(z) = \sum_{n \geq 0} a_n z^n, \tag{A.47}$$

where z is a dummy symbol (or a complex variable). The importance of generating functions is that the function $A(z)$ is a representation of the whole sequence $\langle a_n \rangle$. By studying this function we can get important information about the sequence, such as asymptotic form of the coefficients.

The most basic generating function is the one generating the constant sequence $(1, 1, 1, \ldots)$. As already shown in Appendix A.2, this function is given by

$$\frac{1}{1-z} = \sum_{n \geq 0} z^n. \tag{A.48}$$

A.3.2 Fibonacci Numbers

As a first non-trivial example of the power of generating functions we consider the famous Fibonacci sequence

$$\langle F_n \rangle = (0, 1, 1, 2, 3, 5, 8, \ldots), \tag{A.49}$$

defined by the recurrence relation

$$F_{n+1} = F_n + F_{n-1}, \qquad (n \geq 1, F_0 = 0, F_1 = 1). \tag{A.50}$$

To find the generating function

$$F(z) = \sum_{n \geq 0} F_n z^n = z + z^2 + 2z^3 + 3z^4 + 5z^5 + 8z^6 + \cdots, \tag{A.51}$$

we multiply the recurrence (A.50) by z^n and sum over $n \geq 1$:

$$\sum_{n \geq 1} F_{n+1} z^n = \sum_{n \geq 1} F_n z^n + \sum_{n \geq 1} F_{n-1} z^n \tag{A.52}$$

$$\frac{F(z) - z}{z} = F(z) + zF(z) \tag{A.53}$$

$$F(z) = \frac{z}{1 - z - z^2}. \tag{A.54}$$

From the basic complex analysis we know that the function $F(z)$ has a *partial fraction expansion* of the form

$$\frac{A}{1 - \alpha z} + \frac{B}{1 - \beta z} = \frac{z}{1 - z - z^2} \tag{A.55}$$

for some numbers α, β, A, B. To find these constants, we write (A.55) as

$$\frac{A}{1 - \alpha z} + \frac{B}{1 - \beta z} = \frac{A(1 - \beta z) + B(1 - \alpha z)}{(1 - \alpha z)(1 - \beta z)} = \frac{z}{1 - z - z^2}. \tag{A.56}$$

For this to hold, we must have

$$(A+B) - (A\beta + B\alpha)z = z \tag{A.57}$$

$$(1-\alpha z)(1-\beta z) = 1 - z - z^2, \tag{A.58}$$

which can be solved straightforwardly as

$$\alpha = \frac{1+\sqrt{5}}{2},\ \beta = \frac{1-\sqrt{5}}{2},\ A = \frac{1}{\sqrt{5}},\ B = -\frac{1}{\sqrt{5}}. \tag{A.59}$$

We can now write

$$F(z) = \frac{A}{1-\alpha z} + \frac{B}{1-\beta z} \tag{A.60}$$

$$= A\sum_{n\geq 0}(\alpha z)^n + B\sum_{n\geq 0}(\beta z)^n \tag{A.61}$$

$$= \sum_{n\geq 0}\left(A\alpha^n + B\beta^n\right) z^n, \tag{A.62}$$

and by plugging the solved values (A.59) into Equation (A.62), we get the closed form solution for the nth Fibonacci number

$$F_n = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right). \tag{A.63}$$

A.3.3 Integer Partitions

Let $q_K(n)$ be the number of partitions of integer n into K parts, i.e., the number of finite non-increasing sequences of non-negative integers $(h_1, \ldots, h_K)$ such that $h_1 + h_2 + \cdots + h_K = n$. For example, $q_3(5) = 5$, since we have

$$5 = 5+0+0 = 4+1+0 = 3+2+0 = 3+1+1 = 2+2+1. \tag{A.64}$$

In this section we want to find the generating function of the numbers $q_K(n)$, i.e.,

$$Q_K(z) = \sum_{n\geq 0} q_K(n) z^n. \tag{A.65}$$

Note that an asymptotic analysis of $Q_K(n)$ is discussed in Appendix A.4.

It is well-known (see, e.g., [3]) that the function generating the numbers $q_K(n)$ is given by

$$Q_K(z) = \frac{1}{1-z}\cdot\frac{1}{1-z^2}\cdot\frac{1}{1-z^3}\cdots\frac{1}{1-z^K}. \tag{A.66}$$

Partition	Star diagram	Conjugate	1.term	2.term	3.term
$5,0,0$	$\star\ \star\ \star\ \star\ \star$	$1,1,1,1,1$	z^5	1	1
$4,1,0$	$\star\ \star\ \star\ \star$ $\star$	$2,1,1,1$	z^3	z^2	1
$3,2,0$	$\star\ \star\ \star$ $\star\ \star$	$2,2,1$	z	z^4	1
$3,1,1$	$\star\ \star\ \star$ $\star$ $\star$	$3,1,1$	z^2	1	z^3
$2,2,1$	$\star\ \star$ $\star\ \star$ $\star$	$3,2$	1	z^2	z^3

Table A.1: Partitions and conjugate partitions of integer 5 into 3 parts.

Intuitively, this result can be understood via an example. Take the above-mentioned case with $n = 5, K = 3$. The generating function is

$$Q_3(z) = \frac{1}{1-z} \cdot \frac{1}{1-z^2} \cdot \frac{1}{1-z^3} \tag{A.67}$$

$$= (1 + z + z^2 + z^3 + \cdots)(1 + z^2 + z^4 + z^6 + \cdots) \cdot (1 + z^3 + z^6 + z^9 + \cdots). \tag{A.68}$$

By the basic definition of generating functions, it is clear that the coefficient of z^5 in the expansion of (A.68) must be $q_3(5) = 5$. To see that this is indeed the case, take a look at Table A.1, where the partitions of 5 into 3 parts are listed. Each partition of n can be represented as a *star diagram* composed of n stars arranged in rows. The number of stars in each row is determined by the elements of the partition. Counting the stars by columns instead of rows, we get the *conjugate partition* of the original partition. Now, each conjugate partition represents a way to get the term z^5 in (A.68). Take, for example, the conjugate partition $(2, 1, 1, 1)$:

1. The number of 1's in the partition is 3, so pick the 3rd order term from $(1 + z + z^2 + z^3 + \cdots)$, i.e., z^3.

2. The number of 2's in the partition is 1, so pick the 1st order term from $(1 + z^2 + z^4 + z^6 + \cdots)$, i.e., z^2.

3. The number of 3's in the partition is 0, so pick the 0th order term from $(1 + z^3 + z^6 + z^9 + \cdots)$, i.e., 1.

We end up with the term $z^3 \cdot z^2 \cdot 1 = z^5$, as desired. The other four partitions are treated similarly. We can therefore conclude that the function $Q_3(z)$ generates numbers $q_3(n)$. The full proof can be found in [3].

A.4 Asymptotic Analysis of Generating Functions

In this appendix we will present methods for finding asymptotic behaviour of a sequence based on the theory of generating functions. For the purposes of this thesis, a powerful method called *singularity analysis* by Flajolet and Odlyzko [16] is especially suitable. Additional sources of information on singularity analysis are [52, 17, 66]. Other asymptotic methods, such as bootstrapping, Tauberian theorems, Darboux's method and the saddle point method are discussed in [10, 20, 19, 71, 66].

Suppose we have found the generating function for a certain sequence of numbers that interests us. The goal of asymptotic analysis is to find a simple function of n which approximates well the values of the sequence when n is large. This can be achieved by analyzing the singularities of the generating function. Suitable asymptotic analysis method is then chosen based on the nature of the singularities.

Especially important is the singularity that is nearest to the origo. As argued in [66], this *dominant singularity* determines the asymptotic growth of the coefficients of the generating function. Therefore, it is only necessary to locate this singularity and analyze the behaviour of the function around it.

A.4.1 Rational Functions

We start the discussion on asymptotic analysis by a relatively simple case of rational generating functions, whose only singularities are poles. Let $f(z)$ be a rational function generating the sequence $\langle a_n \rangle$. Suppose $f(z)$ is analytic at zero and has poles at points $p_1, p_2, \ldots, p_m$. Then there exists m polynomials $(P_1, \ldots, P_m)$ such that exactly

$$a_n = [z^n] f(z) = \sum_{j=1}^{m} P_j(n) p_i^{-n}. \tag{A.69}$$

Furthermore, the degree of P_j is equal to the order of the pole at p_j minus one. In particular, a single pole only contributes a constant term to (A.69). This theorem is proved in, e.g., [66]. In practice, the polynomials P_j can be found via residue calculus.

To illustrate the use of (A.69), let us consider a version of the classic money changing problem: in how many ways can one pay an amount of n cents using only coins of 1, 2 and 5 cents? Let m_n denote this number. To solve the problem, we need to find the generating function $m(z)$ for the sequence $\langle m_n \rangle$. The money changing problem is closely related to the counting of integer partitions discussed in Appendix A.3. Using similar arguments, it is easy to see that the generating function is given by

$$m(z) = \frac{1}{(1-z)(1-z^2)(1-z^5)}, \tag{A.70}$$

which is a rational function and analytic at zero, so (A.69) applies.

The first step is to find the poles of (A.70). From the complex root discussion of Appendix A.1, we have:

- The only pole of $(1-z)$ is 1.
- The poles of $(1-z^2)$ are $1, -1$.
- The poles of $(1-z^5)$ are:
  - 1
  - $\cos\frac{2\pi}{5} + i\sin\frac{2\pi}{5}$
  - $\cos\frac{4\pi}{5} + i\sin\frac{4\pi}{5}$
  - $\cos\frac{6\pi}{5} + i\sin\frac{6\pi}{5}$
  - $\cos\frac{8\pi}{5} + i\sin\frac{8\pi}{5}$.

Thus, the function $m(z)$ has a triple pole at $z = 1$ and several single poles. We choose here to ignore the single poles, since they only contribute a constant term to (A.69).

By the Laurent's theorem presented in Appendix A.1, we know that $m(z)$ has a Laurent expansion at $z = 1$,

$$m(z) = \frac{a_{-3}}{(z-1)^3} + \frac{a_{-2}}{(z-1)^2} + \frac{a_{-1}}{(z-1)} + \sum_{n\geq 0} a_n (z-1)^n. \tag{A.71}$$

The coefficients a_n can be found via basic residue calculus. By the coefficient formula (A.13),

$$a_{-3} = \frac{1}{2\pi i} \oint_C \frac{(z-1)^2}{(1-z)(1-z^2)(1-z^5)} dz \tag{A.72}$$

$$= \frac{1}{2\pi i} \oint_C \frac{1}{(1+z)(1-z)(1+z+z^2+z^3+z^4)} dz, \tag{A.73}$$

which is by the residue theorem (A.15)

$$a_{-3} = \operatorname*{Res}_{z=1} \frac{1}{(1+z)(1-z)(1+z+z^2+z^3+z^4)} \tag{A.74}$$

$$= \lim_{z\to 1} \frac{z-1}{(1+z)(1-z)(1+z+z^2+z^3+z^4)} \tag{A.75}$$

$$= \lim_{z\to 1} \frac{-1}{(1+z)(1+z+z^2+z^3+z^4)} \tag{A.76}$$

$$= -\frac{1}{10}. \tag{A.77}$$

Similarly we can calculate that $a_{-2} = 1/4$ and $a_{-1} = -13/40$. The Laurent expansion is then

$$m(z) = -\frac{1}{10(z-1)^3} + \frac{1}{4(z-1)^2} - \frac{13}{40(z-1)} + \sum_{n\geq 0} a_n (z-1)^n \tag{A.78}$$

$$= \frac{1}{10(1-z)^3} + \frac{1}{4(1-z)^2} + \frac{13}{40(1-z)} + \sum_{n\geq 0} a_n (z-1)^n. \tag{A.79}$$

To extract the nth coefficient from the expansion (A.79), we need the following basic combinatoric result (see, e.g., [71])

$$[z^n] \frac{1}{(1-z)^{k+1}} = \binom{n+k}{n}, \tag{A.80}$$

so (see also Table A.2)

$$[z^n] \frac{1}{(1-z)^3} = \binom{n+2}{n} = \frac{1}{2}n^2 + \frac{3}{2}n + 1, \tag{A.81}$$

$$[z^n] \frac{1}{(1-z)^2} = \binom{n+1}{n} = n + 1. \tag{A.82}$$

Now we get the asymptotics for the money changing problem,

$$m_n \sim \frac{1}{10}\left(\frac{1}{2}n^2 + \frac{3}{2}n + 1\right) + \frac{1}{4}(n+1) + \mathcal{O}(1) \tag{A.83}$$

$$= \frac{1}{20}n^2 + \frac{8}{20}n + \mathcal{O}(1). \tag{A.84}$$

To assess the accuracy of the approximation (A.84), we used Maple to calculate the full expansion of the generating function (A.70) therefore obtaining the exact sequence $\langle m_n \rangle$. The comparison of the exact and asymptotic values is given in Figures A.3 and A.4. Clearly, the approximation works very well.

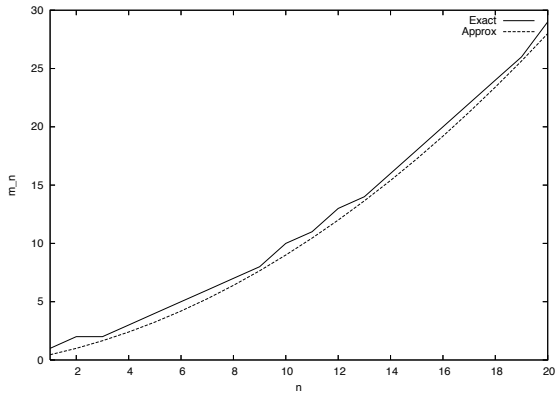

Figure A.3: The comparison of the exact and approximative solutions for the money changing problem with $n = 1, \ldots, 20$.

A.4.2 Asymptotics of Integer Partitions

In this section we briefly discuss the asymptotic analysis of the integer partition generating function (A.66) introduced in Appendix A.3,

$$Q_K(z) = \frac{1}{1-z} \cdot \frac{1}{1-z^2} \cdot \frac{1}{1-z^3} \cdots \frac{1}{1-z^K}. \tag{A.85}$$

Clearly, this function has a pole of order K at $z = 1$. From the discussion of the previous section we know that the highest order pole dominates the asymptotics of rational generating functions. Furthermore, by Equation (A.69) a pole of order K contributes a term of degree $K - 1$. Thus, we can conclude that the number of partitions of an integer n into K parts is $\mathcal{O}\left(n^{K-1}\right)$, i.e., asymptotically the same as the number of compositions.

A.4.3 Algebraic-Logarithmic Functions: The Singularity Analysis

A very general and powerful asymptotic method called *singularity analysis* was introduced in [16]. In its most general form it allows to find asymptotics for *algebraic-logarithmic* functions of the form

$$(1-z)^{-\alpha} \left(\frac{1}{z} \log \frac{1}{1-z} \right)^{\beta}, \tag{A.86}$$

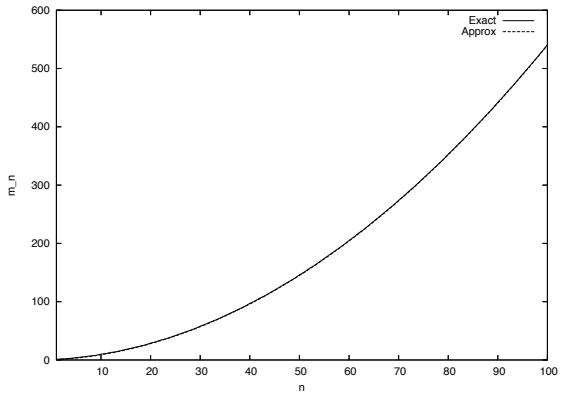

Figure A.4: The comparison of the exact and approximative solutions for the money changing problem for $n = 1, \ldots, 100$.

for real numbers α, β. For the purposes of the other parts of this thesis, however, the following special version is more appropriate: Let $\alpha \neq 0, -1, -2, \ldots$. Then the coefficient of z^n in $(1-z)^{-\alpha}$ is given by

$$[z^n](1-z)^{-\alpha} \sim \frac{n^{\alpha-1}}{\Gamma(\alpha)} \left(1 + \sum_{k=1}^{\infty} \frac{e_k(\alpha)}{n^k}\right), \tag{A.87}$$

where $e_k(\alpha)$ is a polynomial in α of degree $2k$. The first few polynomials are given by

$$e_1(\alpha) = \frac{\alpha(\alpha-1)}{2} \tag{A.88}$$

$$e_2(\alpha) = \frac{\alpha(\alpha-1)(\alpha-2)(3\alpha-1)}{24} \tag{A.89}$$

$$e_3(\alpha) = \frac{\alpha^2(\alpha-1)^2(\alpha-2)(\alpha-3)}{48}. \tag{A.90}$$

The exact definition of these polynomials is complicated but can be found in [66].

To illustrate the use of (A.87), we show how to calculate the asymptotic form for the coefficients of $(1-az)^{-1/2}$, where a is a constant. Firstly, we notice a simple fact that

$$[z^n](1-az)^{-\alpha} = a^n[z^n](1-z)^{-\alpha}. \tag{A.91}$$

Function	Coefficients
$(1-z)^{3/2}$	$\frac{1}{\sqrt{\pi n^5}}\left(\frac{3}{4}+\frac{45}{32n}+\frac{1155}{512n^2}+\mathcal{O}\left(\frac{1}{n^3}\right)\right)$
$(1-z)$	0
$(1-z)^{1/2}$	$-\frac{1}{\sqrt{\pi n^3}}\left(\frac{1}{2}+\frac{3}{16n}+\frac{25}{256n^2}+\mathcal{O}\left(\frac{1}{n^3}\right)\right)$
1	0
$(1-z)^{-1/2}$	$\frac{1}{\sqrt{\pi n}}\left(1-\frac{1}{8n}+\frac{1}{128n^2}+\mathcal{O}\left(\frac{1}{n^3}\right)\right)$
$(1-z)^{-1}$	1
$(1-z)^{-3/2}$	$\sqrt{\frac{n}{\pi}}\left(2+\frac{3}{4n}-\frac{7}{64n^2}+\mathcal{O}\left(\frac{1}{n^3}\right)\right)$
$(1-z)^{-2}$	$n+1$
$(1-z)^{-3}$	$\frac{1}{2}n^2+\frac{3}{2}n+1$
$(1-z)^{-4}$	$\frac{1}{6}n^3+n^2+\frac{11}{6}n+1$

Table A.2: Some commonly encountered functions and the asymptotic form of their coefficients.

The value of α in our example is $1/2$, so

$$[z^n](1-az)^{-1/2} \sim a^n \cdot \frac{n^{-1/2}}{\Gamma(1/2)}\left[1+\frac{(1/2)(-1/2)}{2n}\right. \tag{A.92}$$
$$\left.+\frac{(1/2)(-1/2)(-3/2)(1/2)}{24n^2}+\mathcal{O}\left(\frac{1}{n^3}\right)\right]$$
$$= a^n \cdot \frac{1}{\sqrt{\pi n}}\left(1-\frac{1}{8n}+\frac{1}{128n^2}+\mathcal{O}\left(\frac{1}{n^3}\right)\right). \tag{A.93}$$

Further examples are listed in Table A.2.

Another very important result of singularity analysis is the following *transfer theorem*: If a generating function $A(z)$ satisfies

$$A(z) = \mathcal{O}\left((1-z)^{-\alpha}\right), \tag{A.94}$$

then

$$[z^n]A(z) = \mathcal{O}\left(n^{\alpha-1}\right). \tag{A.95}$$

The same holds for the $o(\cdot)$-functions. Comparing the transfer theorem (A.95) to Equation (A.87), we can see that it is actually very intuitive.

We finalize this section by summarizing the method of singularity analysis into the following recipe:

1. Find the generating function $A(z)$ for the sequence we are interested in.
2. Find the dominant singularity of $A(z)$.
3. Expand $A(z)$ into series around the dominant singularity.
4. Apply Theorems (A.87) and (A.95) to get the asymptotic form for the coefficients.

A highly non-trivial example of using this recipe is presented in Appendix B.

Chapter B

The Szpankowski Approximation

In this appendix we will first derive a generating function for the sequence of multinomial regret terms. This function is used twice in the other parts of this thesis: The elegant recursion formula for exact NML computation in Section 3.1.1 and the accurate Szpankowski approximation in Section 3.1.2 are based on this generating function. Secondly, we give full derivation of the Szpankowski approximation.

B.1 The Regret Generating Function

Let us start with the sequence $\langle n^n/n! \rangle$. As in [66], we denote the function generating this sequence by $B(z)$. Unfortunately, there is no closed-form formula for $B(z)$. As we will see later, this function is nevertheless suitable for our purposes. The connection between $B(z)$ and the multinomial regret

terms can be seen by squaring $B(z)$,

$$B^2(z) = \left(\sum_{r\geq 0} \frac{r^r}{r!} z^r\right) \cdot \left(\sum_{s\geq 0} \frac{s^s}{s!} z^s\right) \tag{B.1}$$

$$= \sum_{r,s\geq 0} \frac{r^r s^s}{r!s!} z^{r+s} \tag{B.2}$$

$$= \sum_{n\geq 0} \left(\sum_{r+s=n} \frac{n^n}{n!} \frac{n!}{r!s!} \frac{r^r s^s}{n^{r+s}}\right) z^n \tag{B.3}$$

$$= \sum_{n\geq 0} \frac{n^n}{n!} \left(\sum_{r+s=n} \frac{n!}{r!s!} \left(\frac{r}{n}\right)^r \left(\frac{s}{n}\right)^s\right) z^n \tag{B.4}$$

$$= \sum_{n\geq 0} \frac{n^n}{n!} \mathcal{C}(\mathcal{M}(2), n) z^n, \tag{B.5}$$

where $\mathcal{M}(2)$ is the multinomial model class with two values. Thus, $B^2(z)$ generates the sequence $\langle \frac{n^n}{n!} \mathcal{C}(\mathcal{M}(2), n)\rangle$. This easily generalizes to

$$B^K(z) = \sum_{n\geq 0} \frac{n^n}{n!} \left[\sum_{h_1+\cdots+h_K=n} \frac{n!}{h_1! \cdots h_K!} \left(\frac{h_1}{n}\right)^{h_1} \cdots \left(\frac{h_k}{n}\right)^{h_k}\right] z^n \tag{B.6}$$

$$= \sum_{n\geq 0} \frac{n^n}{n!} \mathcal{C}(\mathcal{M}(K), n) z^n, \tag{B.7}$$

generating the sequence $\langle \frac{n^n}{n!} \mathcal{C}(\mathcal{M}(K), n)\rangle$. Note that to be precise, the function $B^K(z)$ is the *tree-like generating function* [66] of the sequence $\langle \mathcal{C}(\mathcal{M}(K), n)\rangle$. For simplicity, however, we just call it the regret generating function.

To make the Equation (B.7) useful, we will derive a relation of $B^K(z)$ and the so-called *Cayley's tree function* $T(z)$ [30, 9], which generates the sequence $\langle n^{n-1}/n!\rangle$, i.e.,

$$T(z) = \sum_{n\geq 1} \frac{n^{n-1}}{n!} z^n, \tag{B.8}$$

as shown in [66]. This sequence counts the *rooted labeled trees*, hence the name of the function. The tree function is defined by the functional equation

$$T(z) = z e^{T(z)}. \tag{B.9}$$

Differentiating and multiplying (B.8) by z, we get

$$zT'(z) = z \cdot \sum_{n \geq 1} \frac{n \cdot n^{n-1}}{n!} z^{n-1} \tag{B.10}$$

$$= \sum_{n \geq 1} \frac{n^n}{n!} z^n \tag{B.11}$$

$$= \sum_{n \geq 0} \frac{n^n}{n!} z^n - 1, \tag{B.12}$$

from which we get

$$B(z) = zT'(z) + 1. \tag{B.13}$$

On the other hand, differentiating the functional equation (B.9) gives

$$T'(z) = e^{T(z)} + ze^{T(z)} \cdot T'(z) \tag{B.14}$$

$$T'(z)(1 - ze^{T(z)}) = e^{T(z)} \tag{B.15}$$

$$zT'(z)(1 - T(z)) = T(z) \tag{B.16}$$

$$zT'(z) = \frac{T(z)}{1 - T(z)}. \tag{B.17}$$

Combining the Equations (B.13) and (B.17), we get

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

and thus

$$B^K(z) = \frac{1}{(1 - T(z))^K}. \tag{B.19}$$

This final form can now applied in NML computation by using the properties of the tree function $T(z)$.

B.2 The Derivation

The proof of the Szpankowski approximation (3.8) was only outlined in [66]. We will now present a full derivation. Our starting point is the regret generating function already discussed in Appendix B.1,

$$B^K(z) = \frac{1}{(1 - T(z))^K} = \sum_{n \geq 0} \frac{n^n}{n!} \mathcal{C}(\mathcal{M}(K), n) z^n. \tag{B.20}$$

To make the presentation easier to follow, the derivation is split into the following steps:

1. Find the dominant singularity of the regret generating function $B^K(z)$.
2. Expand the inverse of the tree function $T(z)$ into series around the dominant singularity point.
3. Invert this series to get the expansion of the tree function.
4. Find the series for $B(z) = 1/(1 - T(z))$.
5. Find the series for $B^K(z)$.
6. Apply the singularity analysis theorem (A.87) term by term.
7. Multiply by $n!/n^n$ to extract the asymptotic form of the regret terms.
8. Take the logarithm to prove (3.8).

Step 1: To get the asymptotic form for the coefficients of (B.20), we need to expand the function $B^K(z)$ around its dominant singularity, i.e., the one nearest to the origo. It is well-known (see, e.g., [9]) that the dominant singularity of $T(z)$ occurs at $z = 1/e$. This point is also the dominant singularity of (B.20), since the zero of the denominator (pole) is also at $z = 1/e$. This can be seen by solving z from the functional equation (B.9),

$$z = F(T) = Te^{-T}, \tag{B.21}$$

and then plugging $T = 1$ into it.

Step 2: Deriving the series expansion for (B.20) is a very non-trivial task, since there is no explicit formula for $B(z)$ or $T(z)$. It turns out that the inverse function $F(T)$ is a good starting point, since it is an entire function (analytic everywhere). To get the expansion of $T(z)$ around $z = 1/e$, we can first expand $F(T)$ around $T = 1$, and then use the series inversion method described in Appendix A.2.5. Since $F(T)$ is entire, its expansion is a simple Taylor series, which can be found by calculating the derivatives of $F(T)$ at $T = 1$. We have

$$\begin{aligned}
F'(T) &= e^{-T} + T \cdot (-e^{-T}) = e^{-T}(1 - T) && \text{(B.22)}\\
F''(T) &= -e^{-T}(1 - T) - e^{-T} = -e^{-T}(2 - T) && \text{(B.23)}\\
F'''(T) &= e^{-T}(2 - T) + e^{-T} = e^{-T}(3 - T) && \text{(B.24)}\\
F''''(T) &= -e^{-T}(3 - T) - e^{-T} = -e^{-T}(4 - T), && \text{(B.25)}
\end{aligned}$$

which leads to

$$F(T) = F(1) + F'(1)(T-1) + \frac{F''(1)}{2!}(T-1)^2 + \frac{F'''(1)}{3!}(T-1)^3 + \frac{F''''(1)}{4!}(T-1)^4 + \cdots \quad \text{(B.26)}$$

$$= 1/e - \frac{1/e}{2}(T-1)^2 + \frac{1/e}{3}(T-1)^3 - \frac{1/e}{8}(T-1)^4 + \cdots \quad \text{(B.27)}$$

$$= 1/e - \frac{1/e}{2}(1-T)^2 - \frac{1/e}{3}(1-T)^3 - \frac{1/e}{8}(1-T)^4 + \cdots . \quad \text{(B.28)}$$

Step 3: Looking at Equation (B.22), we can see that the first derivative vanishes at $T = 1$. As suggested in Appendix A.2.5, this unfortunately means that inverting the series (B.28) is not straightforward. Intuitively, this complication can be understood via Figure B.1, where the function $F(T)$ is plotted near the point $T = 1$ (in real number space). Clearly, $F(T)$ is non-monotonic in every neighborhood of $T = 1$, and the

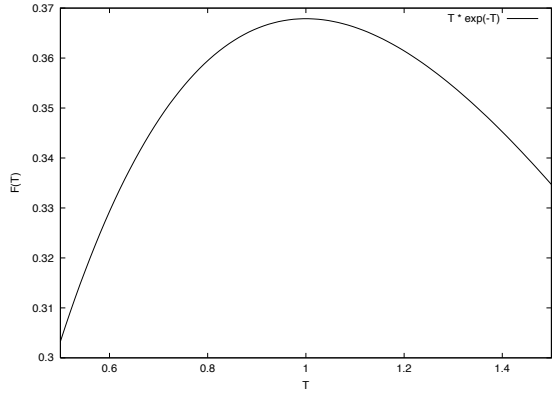

Figure B.1: Plot of $F(T) = Te^{-T}$ around $T = 1$.

inverse function thus multiple-valued. It follows that the expansion of $T(z)$ around point $z = 1/e$ must also contain multiple-valued terms. As we will soon see, this is indeed the case: the inverted series will be a *Puiseux series* with fractional power terms. To read more about Puiseux series, see Appendix A.1.7.

To find the inverse of (B.28), we can use a theorem from [14], which classifies series expansions into four types of systematic patterns based on

the first few terms of the series. With the terminology of [14], our series falls into category "Type II" with the order parameter β set to 2 (see also Appendix A.1.7). For this category, the series inversion is performed by starting with variable transformations

$$v = 1 - T \tag{B.29}$$

$$w = (1/e - F(T))^{1/\beta} = (1/e - z)^{1/2}, \tag{B.30}$$

and then examining the function

$$w = A(v) = (1/e - F(T))^{1/2} \tag{B.31}$$

$$= (f_2 v^2 + f_3 v^3 + f_4 v^4 + \cdots)^{1/2}, \tag{B.32}$$

where, from (B.28),

$$f_2 = \frac{1/e}{2}, \quad f_3 = \frac{1/e}{3}, \quad f_4 = \frac{1/e}{8}. \tag{B.33}$$

Next we need to find the series expansion for function $A(v)$, i.e., coefficients s_n such that

$$\left(f_2 v^2 + f_3 v^3 + f_4 v^4 + \cdots\right)^{1/2} = s_1 v + s_2 v^2 + s_3 v^3 + \cdots . \tag{B.34}$$

It is easy to prove (see also Figure B.1) that $1/e - F(T) \geq 0$ for all $T \in \mathbb{R}$, from which it follows that we can square both sides of (B.34)

$$f_2 v^2 + f_3 v^3 + f_4 v^4 + \cdots = \left(s_1 v + s_2 v^2 + s_3 v^3 + \cdots\right)^2 \tag{B.35}$$

$$= s_1^2 v^2 + 2 s_1 s_2 v^3 + (2 s_1 s_3 + s_2^2) v^4 + \cdots , \tag{B.36}$$

and by coefficient comparison

$$s_1^2 = f_2, \quad s_1 = \sqrt{f_2} \tag{B.37}$$

$$2 s_1 s_2 = f_3, \quad s_2 = \frac{f_3}{2 s_1} = \frac{f_3}{2\sqrt{f_2}} \tag{B.38}$$

$$2 s_1 s_3 + s_2^2 = f_4, \quad s_3 = \frac{f_4 - s_2^2}{2 s_1} = \frac{4 f_2 f_4 - f_3^2}{8 f_2^{3/2}}. \tag{B.39}$$

The function $A(v)$ can now be written as

$$A(v) = \sqrt{f_2} v + \frac{f_3}{2\sqrt{f_2}} v^2 + \frac{4 f_2 f_4 - f_3^2}{8 f_2^{3/2}} v^3 + \cdots , \tag{B.40}$$

from which we can finally see the idea behind the transformations (B.29) and (B.30). That is, series (B.40) is an ordinary power series with zero constant coefficient therefore having a well-defined inverse, say,

$$v = D(w) = d_1 w + d_2 w^2 + d_3 w^3 + \cdots , \tag{B.41}$$

where the coefficients d_n are given by (see Appendix A.2.5)

$$d_1 = \frac{1}{s_1} = \frac{1}{\sqrt{f_2}} = \sqrt{2e} \tag{B.42}$$

$$d_2 = -\frac{s_2}{s_1^3} = -\frac{f_3}{2f_2^2} = -\frac{2}{3}e \tag{B.43}$$

$$d_3 = \frac{2s_2^2}{s_1^5} - \frac{s_3}{s_1^4} = \frac{5f_3^2 - 4f_2 f_4}{8f_2^{7/2}} = \frac{11\sqrt{2}}{36}e^{3/2}. \tag{B.44}$$

Transforming back to original variables gives the series expansion for the tree function

$$T(z) = 1 - D(w) \tag{B.45}$$

$$= 1 - \sqrt{2e}(1/e - z)^{1/2} + \frac{2}{3}e(1/e - z) - \frac{11\sqrt{2}}{36}e^{3/2}(1/e - z)^{3/2} + \cdots , \tag{B.46}$$

which can be further written as

$$T(z) = 1 - \sqrt{2}(1 - ez)^{1/2} + \frac{2}{3}(1 - ez) - \frac{11\sqrt{2}}{36}(1 - ez)^{3/2} + \cdots . \tag{B.47}$$

This final form makes is more convenient to apply singularity analysis in Step 6.

Step 4: After deriving the expansion for $T(z)$, the next task is to find series for

$$B(z) = \frac{1}{1 - T(z)}, \tag{B.48}$$

i.e., the reciprocal series of

$$1 - T(z) = \sqrt{2}(1 - ez)^{1/2} - \frac{2}{3}(1 - ez) + \frac{11\sqrt{2}}{36}(1 - ez)^{3/2} + \cdots . \tag{B.49}$$

It is clear that the reciprocal is of the form

$$B(z) = a(1 - ez)^{-1/2} + b + c(1 - ez)^{1/2} + \cdots , \tag{B.50}$$

for some numbers $(a, b, c, \ldots)$. By the definition of the reciprocal series, we must then have

$$B(z)(1-T(z)) = \left[a(1-ez)^{-1/2} + b + c(1-ez)^{1/2} + \cdots\right] \cdot \left[\sqrt{2}(1-ez)^{1/2} - \frac{2}{3}(1-ez) + \frac{11\sqrt{2}}{36}(1-ez)^{3/2} + \cdots\right] \equiv 1, \quad \text{(B.51)}$$

i.e., the trivial sequence $(1, 0, 0, \ldots)$. The coefficients $(a, b, c, \ldots)$ can be calculated by comparing coefficients

$$\sqrt{2}a = 1, \quad a = \frac{1}{\sqrt{2}} \quad \text{(B.52)}$$

$$-\frac{2}{3}a + \sqrt{2}b = 0, \quad b = \frac{2}{3\sqrt{2}}a = \frac{1}{3} \quad \text{(B.53)}$$

$$\frac{11\sqrt{2}}{36}a - \frac{2}{3}b + \sqrt{2}c = 0, \quad c = -\frac{11}{36}a + \frac{2}{3\sqrt{2}}b = -\frac{\sqrt{2}}{24}, \quad \text{(B.54)}$$

and thus we get the series expansion

$$B(z) = \frac{1}{\sqrt{2}}(1-ez)^{-1/2} + \frac{1}{3} - \frac{\sqrt{2}}{24}(1-ez)^{1/2} + \cdots. \quad \text{(B.55)}$$

Step 5: The final step for deriving the series expansion for the regret generating function (B.20) is to expand

$$B^K(z) = \frac{1}{(1-T(z))^K} = \left(\frac{1}{\sqrt{2}}(1-ez)^{-1/2} + \frac{1}{3} - \frac{\sqrt{2}}{24}(1-ez)^{1/2} + \cdots\right)^K. \quad \text{(B.56)}$$

The first term of this series, i.e., the one with the smallest exponent, is obtained by raising the first term of (B.56) into Kth power

$$\left(\frac{1}{\sqrt{2}}(1-ez)^{-1/2}\right)^K = \left(\frac{1}{\sqrt{2}}\right)^K (1-ez)^{-K/2} = \frac{1}{2^{K/2}}(1-ez)^{-K/2}. \quad \text{(B.57)}$$

To get the next term we raise the first term of (B.56) into $(K-1)$th power and then multiply by the second term. There are K different ways to choose the second term, which gives

$$K \cdot \left(\frac{1}{\sqrt{2}}\right)^{K-1} \cdot \frac{1}{3} \cdot (1-ez)^{-\frac{K}{2}+\frac{1}{2}} = \frac{K}{3 \cdot 2^{\frac{K}{2}-\frac{1}{2}}}(1-ez)^{-\frac{K}{2}+\frac{1}{2}}. \quad \text{(B.58)}$$

For the third term, we need to consider two cases:

1. Raise the first term of (B.56) into $(K-1)$th power and then multiply by the third term. The third term can be chosen in K different ways.

2. Raise the first term of (B.56) into $(K-2)$th power and then multiply by the square of the second term. We have $\binom{K}{2} = K(K-1)/2$ ways to do that.

Thus, the third term of $B^K(z)$ is

$$\left[K \cdot \left(\frac{1}{\sqrt{2}}\right)^{K-1} \cdot \frac{-\sqrt{2}}{24} + \frac{K(K-1)}{2} \cdot \left(\frac{1}{\sqrt{2}}\right)^{K-2} \cdot \left(\frac{1}{3}\right)^2\right] \cdot (1-ez)^{-\frac{K}{2}+1} = \frac{4K(K-1)-3K}{36 \cdot 2^{K/2}}(1-ez)^{-\frac{K}{2}+1}. \quad \text{(B.59)}$$

As we will soon see, it is not necessary to calculate more terms. The series expansion for the regret generating function is now

$$B^K(z) = \frac{1}{2^{K/2}}(1-ez)^{-K/2} + \frac{K}{3 \cdot 2^{\frac{K}{2}-\frac{1}{2}}}(1-ez)^{-\frac{K}{2}+\frac{1}{2}} + \frac{4K(K-1)-3K}{36 \cdot 2^{K/2}}(1-ez)^{-\frac{K}{2}+1} + \cdots . \quad \text{(B.60)}$$

Step 6: We are now ready to apply the singularity analysis theorem (A.87) to series (B.60). Proceeding term by term basis,

$$[z^n]\left(\frac{1}{2^{K/2}}(1-ez)^{-K/2}\right) \sim \quad \text{(B.61)}$$
$$e^n \cdot \frac{n^{\frac{K}{2}-1}}{2^{K/2} \cdot \Gamma(K/2)}\left(1 + \frac{K(K-1)}{2n} + \mathcal{O}\left(1/n^2\right)\right)$$
$$[z^n]\left(\frac{K}{3 \cdot 2^{\frac{K}{2}-\frac{1}{2}}}(1-ez)^{-\frac{K}{2}+\frac{1}{2}}\right) \sim \quad \text{(B.62)}$$
$$e^n \cdot \frac{K \cdot n^{\frac{K}{2}-\frac{3}{2}}}{3 \cdot 2^{\frac{K}{2}-\frac{1}{2}} \cdot \Gamma(\frac{K}{2}-\frac{1}{2})}\left(1 + \frac{K(K-1)}{2n} + \mathcal{O}\left(1/n^2\right)\right)$$
$$[z^n]\left(\frac{4K(K-1)-3K}{36 \cdot 2^{K/2}}(1-ez)^{-\frac{K}{2}+1}\right) \sim \quad \text{(B.63)}$$
$$e^n \cdot \frac{(4K(K-1)-3K) \cdot n^{\frac{K}{2}-2}}{36 \cdot 2^{K/2} \cdot \Gamma(\frac{K}{2}-1)}\left(1 + \frac{K(K-1)}{2n} + \mathcal{O}\left(1/n^2\right)\right).$$

After some tedious algebra we get the asymptotic form for the nth coefficient of the regret generating function:

$$[z^n]B^K(z) \sim e^n \cdot \left[\frac{1}{2^{K/2} \cdot \Gamma(K/2)} \cdot n^{\frac{K}{2}-1} + \frac{K}{2^{\frac{K}{2}-\frac{1}{2}} \cdot 3\Gamma(\frac{K}{2}-\frac{1}{2})} \cdot n^{\frac{K}{2}-\frac{3}{2}} + \frac{K(K-2)(2K+1)}{2^{K/2} \cdot 36\Gamma(K/2)} \cdot n^{\frac{K}{2}-2} + \mathcal{O}\left(n^{\frac{K}{2}-\frac{5}{2}}\right) \right]. \quad \text{(B.64)}$$

Step 7: To extract the asymptotic form of the terms $\mathcal{C}(\mathcal{M}(K), n)$, we need to multiply Equation (B.64) by $n!/n^n$. By the celebrated Stirling's formula,

$$\frac{n!}{n^n} = \sqrt{2\pi n} \cdot e^{-n} \left(1 + \frac{1}{12n} + \mathcal{O}\left(\frac{1}{n^2}\right)\right), \quad \text{(B.65)}$$

which nicely cancels the e^n term in (B.64). Multiplying (B.64) by (B.65) gives after simplifications

$$\begin{aligned} \mathcal{C}(\mathcal{M}(K), n) &\sim \left(\frac{n}{2}\right)^{\frac{K-1}{2}} \cdot \frac{\sqrt{\pi}}{\Gamma(K/2)} \left[1 + \frac{\sqrt{2}K \cdot \Gamma(K/2)}{3\Gamma(\frac{K}{2}-\frac{1}{2})} \cdot \frac{1}{\sqrt{n}} \right. \quad \text{(B.66)} \\ &\quad \left. + \frac{K(K-2)(2K+1)}{36} \cdot \frac{1}{n} + \mathcal{O}\left(\frac{1}{n^{3/2}}\right)\right] \\ &\quad \cdot \left[1 + \frac{1}{12n} + \mathcal{O}\left(\frac{1}{n^2}\right)\right] \\ &= \left(\frac{n}{2}\right)^{\frac{K-1}{2}} \cdot \frac{\sqrt{\pi}}{\Gamma(K/2)} \left[1 + \frac{\sqrt{2}K \cdot \Gamma(K/2)}{3\Gamma(\frac{K}{2}-\frac{1}{2})} \cdot \frac{1}{\sqrt{n}} \right. \quad \text{(B.67)} \\ &\quad \left. + \frac{3 + K(K-2)(2K+1)}{36} \cdot \frac{1}{n} + \mathcal{O}\left(\frac{1}{n^{3/2}}\right)\right]. \end{aligned}$$

Step 8: The final step is to take the logarithm of (B.67). Consider the standard Taylor series of the (natural) logarithm function

$$\log(1+z) = z - \frac{z^2}{2} + \frac{z^3}{3} + \cdots. \quad \text{(B.68)}$$

Plugging

$$z = \frac{a}{\sqrt{n}} + \frac{b}{n} + \mathcal{O}\left(\frac{1}{n^{3/2}}\right) \quad \text{(B.69)}$$

into series (B.68) gives

$$\log\left[1+\frac{a}{\sqrt{n}}+\frac{b}{n}+\mathcal{O}\left(\frac{1}{n^{3/2}}\right)\right]=\frac{a}{\sqrt{n}}+\frac{b}{n}-\frac{1}{2}\left[\frac{a}{\sqrt{n}}+\frac{b}{n}+\mathcal{O}\left(\frac{1}{n^{3/2}}\right)\right]^2 \tag{B.70}$$

$$=\frac{a}{\sqrt{n}}+(b-\frac{1}{2}a^2)\cdot\frac{1}{n}+\mathcal{O}\left(\frac{1}{n^{3/2}}\right), \tag{B.71}$$

for numbers a, b. By applying (B.71) to (B.67) we get the asymptotic formula for the multinomial regret terms:

$$\log\mathcal{C}(\mathcal{M}(K),n)=\frac{K-1}{2}\log\frac{n}{2}+\log\frac{\sqrt{\pi}}{\Gamma(K/2)}+\frac{\sqrt{2}K\cdot\Gamma(K/2)}{3\Gamma(\frac{K}{2}-\frac{1}{2})}\cdot\frac{1}{\sqrt{n}} \tag{B.72}$$

$$+\left(\frac{3+K(K-2)(2K+1)}{36}-\frac{\Gamma^2(K/2)\cdot K^2}{9\Gamma^2\left(\frac{K}{2}-\frac{1}{2}\right)}\right)\cdot\frac{1}{n} \tag{B.73}$$

$$+\mathcal{O}\left(\frac{1}{n^{3/2}}\right).$$

The proof of (3.8) follows trivially.

An important thing to notice is that in all the steps of the derivation we could have calculated an arbitrary number of terms for the series expansions. It follows that the derivation does not limit the accuracy of the final result. However, as shown in Section 3.1.3, $\mathcal{O}\left(1/n^{3/2}\right)$ is accurate enough for practical purposes.

References

[1] M. Abramowitz and I. Stegun, editors. *Handbook of Mathematical Functions*. Dover Publications, Inc., New York, 1970.

[2] A. Asuncion and D. Newman. UCI machine learning repository, 2007. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[3] V. Balakrishnan. *Schaum's Outline of Theory and Problems of Combinatorics*. McGraw-Hill, 1995.

[4] V. Balasubramanian. MDL, Bayesian inference, and the geometry of the space of probability distributions. In P. Grünwald, I. Myung, and M. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*, pages 81–98. The MIT Press, 2006.

[5] A. Barron, J. Rissanen, and B. Yu. The minimum description principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, October 1998.

[6] L. Birge and Y. Rozenholc. How many bins should be put in a regular histogram. Prepublication no 721, Laboratoire de Probabilites et Modeles Aleatoires, CNRS-UMR 7599, Universite Paris VI & VII, April 2002.

[7] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman. Autoclass: A Bayesian classification system. In *Proceedings of the Fifth International Conference on Machine Learning*, pages 54–64, Ann Arbor, June 1988.

[8] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[9] R. Corless, G. Gonnet, D. Hare, D. Jeffrey, and D. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5:329–359, 1996.

[10] N. De Bruijn. *Asymptotic Methods in Analysis*. Dover Publications, Inc., New York, 1981.

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

[12] E. Elovaara and P. Myllymäki. MDL-based attribute models in naive Bayes classification. In *Workshop on Information Theoretic Methods in Science and Engineering (WITMSE)*, Tampere, Finland, 2009.

[13] B. Everitt and D. Hand. *Finite Mixture Distributions*. Chapman and Hall, London, 1981.

[14] B. Fabinojas. Laplace's method on a computer algebra system with an application to the real valued modified Bessel functions. *Journal of Computational and Applied Mathematics*, 146:323–342, 2002.

[15] W. Feller. *An Introduction to Probability Theory and Its Applications*. John Wiley & Sons, 3rd edition, 1968.

[16] P. Flajolet and A. Odlyzko. Singularity analysis of generating functions. *SIAM Journal on Discrete Mathematics*, 3(2):216–240, 1990.

[17] P. Flajolet and R. Sedgewick. The average case analysis of algorithms : Complex asymptotics and generating functions. Technical Report RR-2026, INRIA, 1993.

[18] C. Fraley and A. E. Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[19] R. Graham, D. Knuth, and O. Patashnik. *Concrete Mathematics (second edition)*. Addison-Wesley, 1994.

[20] D. Greene and D. Knuth. *Mathematics for the Analysis of Algorithms*. Birkhäuser Boston, 1982.

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

[22] P. Grünwald, P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. Minimum encoding approaches for predictive modeling. In G. Cooper and S. Moral, editors, *Proceedings of the 14th International Conference on Uncertainty in Artificial Intelligence (UAI'98)*, pages 183–192,

Madison, WI, July 1998. Morgan Kaufmann Publishers, San Francisco, CA.

[23] P. Hall and E. Hannan. On stochastic complexity and nonparametric density estimation. *Biometrika*, 75(4):705–714, 1988.

[24] D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Advanced Technology Division, One Microsoft Way, Redmond, WA 98052, 1996.

[25] P. Henrici. Automatic computations with power series. *Journal of the ACM*, 3(1):11–15, January 1956.

[26] P. Henrici. *Applied and Computational Complex Analysis, Vols. 1–3.* John Wiley & Sons, New York, 1977.

[27] D. Knuth. *The Art of Computer Programming, vol. 1 / Fundamental Algorithms (third edition).* Addison-Wesley, 1997.

[28] D. Knuth. *The Art of Computer Programming, vol. 2 / Seminumerical Algorithms (third edition).* Addison-Wesley, 1998.

[29] D. Knuth. *The Art of Computer Programming, vol. 3 / Sorting and Searching (second edition).* Addison-Wesley, 1998.

[30] D. Knuth and B. Pittel. A recurrence related to trees. *Proceedings of the American Mathematical Society*, 105(2):335–349, 1989.

[31] 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.

[32] P. Kontkanen, W. Buntine, P. Myllymäki, J. Rissanen, and H. Tirri. Efficient computation of stochastic complexity. In C. Bishop and B. Frey, editors, *Proceedings of the Ninth International Conference on Artificial Intelligence and Statistics*, pages 233–238. Society for Artificial Intelligence and Statistics, 2003.

[33] P. Kontkanen, J. Lahtinen, P. Myllymäki, T. Silander, and H. Tirri. Supervised model-based visualization of high-dimensional data. *Intelligent Data Analysis*, 4:213–227, 2000.

[34] P. Kontkanen and 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.

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

[36] P. Kontkanen and P. Myllymäki. MDL histogram density estimation. In M. Meila and S. Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, March 2007.

[37] P. Kontkanen and P. Myllymäki. An empirical comparison of NML clustering algorithms. In *Proceedings of the 2008 International Conference on Information Theory and Statistical Learning (ITSL-08)*, 2008.

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

[39] P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. On Bayesian case matching. In B. Smyth and P. Cunningham, editors, *Advances in Case-Based Reasoning, Proceedings of the 4th European Workshop (EWCBR-98)*, volume 1488 of *Lecture Notes in Artificial Intelligence*, pages 13–24. Springer-Verlag, 1998.

[40] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald. On predictive distributions and Bayesian networks. *Statistics and Computing*, 10:39–54, 2000.

[41] P. Kontkanen, P. Myllymäki, and H. Tirri. Constructing Bayesian finite mixture models by the EM algorithm. Technical Report NC-TR-97-003, ESPRIT Working Group on Neural and Computational Learning (NeuroCOLT), 1996.

[42] P. Kontkanen, H. Wettig, and P. Myllymäki. NML computation algorithms for tree-structured multinomial Bayesian networks. *EURASIP Journal on Bioinformatics and Systems Biology*, Article ID 90947, 2007.

[43] G. Korodi and I. Tabus. An efficient normalized maximum likelihood algorithm for DNA sequence compression. *ACM Trans. Inf. Syst.*, 23(1):3–34, 2005.

[44] G. McLachlan, editor. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

[45] M. Meila and D. Heckerman. An experimental comparison of several clustering and initialization methods. In G. F. Cooper and S. Moral, editors, *UAI'98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 386–395, 1998.

[46] T. Mononen and P. Myllymäki. Fast NML computation for Naive Bayes models. In V. Corruble, M. Takeda, and E. Suzuki, editors, *Proceedings of the Tenth International Conference on Discovery Science*, October 2007.

[47] T. Mononen and P. Myllymäki. Computing the multinomial stochastic complexity in sub-linear time. In *Proceedings of European Workshop on Probabilistic Graphical Models (PGM'08)*, pages 209–216, 2008.

[48] T. Mononen and P. Myllymäki. Computing the NML for Bayesian forests via matrices and generating polynomials. In *IEEE Information Theory Workshop*, Porto, Portugal, May 2008.

[49] T. Mononen and P. Myllymäki. On recurrence formulas for computing the stochastic complexity. In *Proceedings of the International Symposium on Information Theory and its Applications*, pages 281–286, Auckland, New Zealand, 2008. IEEE.

[50] T. Mononen and P. Myllymäki. On the multinomial stochastic complexity and its connection to the birthday problem. In *International Conference on Information Theory and Statistical Learning*, Las Vegas, NV, July 2008.

[51] T. Needham. *Visual Complex Analysis*. Oxford University Press, 1997.

[52] A. Odlyzko. Asymptotic enumeration methods. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, volume 2, pages 1063–1229. North-Holland, Amsterdam, 1995.

[53] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:445–471, 1978.

[54] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society*, 49(3):223–239 and 252–265, 1987.

[55] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Company, New Jersey, 1989.

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

[57] 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, July 2001.

[58] J. Rissanen. *Information and Complexity in Statistical Modeling*. Springer, 2007.

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

[60] T. Roos, P. Myllymäki, and H. Tirri. On the behavior of MDL denoising. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 309–316, 2005.

[61] T. Roos, T. Silander, P. Kontkanen, and P. Myllymäki. Bayesian network structure learning using factorized NML universal models. In *Information Theory and Applications Workshop*, San Diego, CA, January 2008.

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

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

[64] P. Smyth. Probabilistic model-based clustering of multivariate and sequential data. In D. Heckerman and J. Whittaker, editors, *Proceedings of the Seventh International Conference on Artificial Intelligence and Statistics*, pages 299–304. Morgan Kaufmann Publishers, 1999.

[65] M. Spiegel. *Schaum's Outline of Theory and Problems of Complex Variables*. McGraw-Hill, 1981.

[66] W. Szpankowski. *Average case analysis of algorithms on sequences*. John Wiley & Sons, 2001.

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

[68] H. Tirri. *Plausible Prediction by Bayesian Inference*. PhD thesis, Report A-1997-1, Department of Computer Science, University of Helsinki, June 1997.

[69] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, New York, 1985.

[70] H. Wettig, P. Kontkanen, and P. Myllymäki. Calculating the normalized maximum likelihood distribution for Bayesian forests. *IADIS International Journal on Computer Science and Information Systems*, 2, October 2007.

[71] H. Wilf. *generatingfunctionology (second edition)*. Academic Press, 1994.

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

[73] B. Yu and T. Speed. Data compression and histograms. *Probab. Theory Relat. Fields*, 92:195–229, 1992.

[74] D. Zill and P. Shanahan. *A First Course in Complex Analysis with Applications*. Jones and Bartlett Publishers, Inc., 2003.

Paper I

P. Kontkanen, W.Buntine, P. Myllymäki, J.Rissanen, H.Tirri

Efficient Computation of Stochastic Complexity

Pp. 181–188 in Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, edited by Christopher M. Bishop and Brendan J. Frey, 2003.

Pp. 181–188 in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, edited by Christopher M. Bishop and Brendan J. Frey. Society for Artificial Intelligence and Statistics, 2003.

Efficient Computation of Stochastic Complexity

Petri Kontkanen, Wray Buntine, Petri Myllymäki, Jorma Rissanen, Henry Tirri
Complex Systems Computation Group (CoSCo), Helsinki Institute for Information Technology (HIIT)
University of Helsinki & Helsinki University of Technology
P.O. Box 9800, FIN-02015 HUT, Finland.
{Firstname}.{Lastname}@hiit.fi

Abstract

Stochastic complexity of a data set is defined as the shortest possible code length for the data obtainable by using some fixed set of models. This measure is of great theoretical and practical importance as a tool for tasks such as model selection or data clustering. Unfortunately, computing the modern version of stochastic complexity, defined as the Normalized Maximum Likelihood (NML) criterion, requires computing a sum with an exponential number of terms. Therefore, in order to be able to apply the stochastic complexity measure in practice, in most cases it has to be approximated. In this paper, we show that for some interesting and important cases with multinomial data sets, the exponentiality can be removed without loss of accuracy. We also introduce a new computationally efficient approximation scheme based on analytic combinatorics and assess its accuracy, together with earlier approximations, by comparing them to the exact form. The results suggest that due to its accuracy and efficiency, the new sharper approximation will be useful for a wide class of problems with discrete data.

1 INTRODUCTION

From the information-theoretic point of view, the most plausible explanation for a phenomenon is the one which can be used for constructing the most effective coding of the observable realizations of the phenomenon. This type of *minimum encoding* explanations can be applied in statistical learning for building realistic domain models, given some sample data. Intuitively speaking, in principle this approach can be argued to produce the best possible model of the problem domain, since in order to be able to produce the most efficient coding of data, one must capture all the regularities present in the domain. Consequently, the minimum encoding approach can be used for constructing a solid theoretical framework for statistical modeling. Similarly, the minimum encoding approach can be used for producing accurate predictions of future events.

The most well-founded theoretical formalization of the intuitively appealing minimum encoding approach is the *Minimum Description Length (MDL)* principle developed by Rissanen (Rissanen, 1978, 1987, 1996). The MDL principle has gone through several evolutionary steps during the last two decades. For example, the early realization of the MDL principle, the two-part code MDL (Rissanen, 1978), takes the same form as the Bayesian BIC criterion (Schwarz, 1978), which has led some people to incorrectly believe that MDL and BIC are equivalent. The latest instantiation of MDL discussed here is *not* directly related to BIC, but to a more evolved formalization described in (Rissanen, 1996). For discussions on the theoretical advantages of this approach, see e.g. (Rissanen, 1996; Barron, Rissanen, & Yu, 1998; Grünwald, 1998; Rissanen, 1999; Xie & Barron, 2000; Rissanen, 2001) and the references therein.

The most important notion of MDL is the *Stochastic Complexity (SC)*, which is defined as the shortest description length of a given data relative to a model class $\mathcal{M}$. Unlike some other approaches, like for example Bayesian methods, the MDL principle does not assume that the model class chosen is correct. It even says that there is no such thing as a true model or model class, which in Bayesian methods is sometimes acknowledged in practice. Furthermore, SC is an objective criterion in the sense that it is not dependent on any prior distribution, it only uses the data at hand[1]. This means that the objectives of the MDL approach are very similar to those behind Bayesian methods with so-called reference priors (Bernardo, 1997), but note, however, that Bernardo himself expresses doubt that a reasonably general notion of "non-informative" pri-

[1]Unlike Bayesian methods, with SC the possible subjective prior information is not used as an explicit part of the theoretical framework, but it is expected to be used implicitly in the selection of the parametric model class discussed in the next section.

ors exists in Bayesian statistics in the multivariate framework (Bernardo, 1997).

It has been shown (see (Clarke & Barron, 1990; Grünwald, 1998)) that the stochastic complexity criterion is asymptotically equivalent to the asymptote of the Bayesian marginal likelihood method with the Jeffreys prior under certain conditions, when the Jeffreys prior also becomes equivalent to the so-called reference priors (Bernardo & Smith, 1994). Nevertheless, with discrete data this equivalence does not hold near the boundary of the parameter space in many models (Chickering & Heckerman, 1997; Xie & Barron, 2000), and in applications such as document or natural language modelling some parameters are expected to lie at the boundary. The implicit use of the Laplace approximation in the Bayesian derivations severely strains the approximation or completely anulls it on the boundaries, as discussed in (Bernardo & Smith, 1994; Bleistein & Handelsman, 1975). Consequently, it can be said that the stochastic complexity approach aims to achieve the goal of objectivity in a way not demonstrated in the Bayesian approach due to technical difficulties.

All this makes the MDL principle theoretically very appealing. However, the applications of the modern, so called Normalized Maximum Likelihood (NML) version of MDL, at least with multinomial data, have been quite rare. This is due to the fact that the definition of SC involves a sum (or integral) over all the possible data matrices of certain length, which are obviously exponential in number. Some applications have been presented for discrete regression (Tabus, Rissanen, & Astola, 2002), linear regression (Barron et al., 1998; Dom, 1996), density estimation (Barron et al., 1998) and segmentation of binary strings (Dom, 1995). In this paper, we will present methods for removing the exponentiality of SC in several important cases involving multinomial (discrete) data. Even these methods are, however, in some cases computationally demanding. Therefore we also present three computationally efficient approximations to SC and instantiate them for the cases mentioned. The approach is similar to our previous work in (Kontkanen, Myllymäki, Silander, & Tirri, 1999), but it was based on an earlier definition of MDL, not on the modern version adopted here. The ability to compute the exact SC gives us a unique opportunity to see how accurate the approximations are. This is important as we firmly believe that the results extend to more complex cases where exact SC is not available.

In Section 2 we first review the MDL principle and discuss how to compute it for a single multinomial variable and a certain multi-dimensional model class. The techniques used in this section are completely new. Section 3 presents the three SC approximations for multinomial data. In Section 4 we study the accuracy of these approximations by comparing them to the exact stochastic complexity. Finally, Section 5 gives the concluding remarks and presents some ideas for future work.

2 STOCHASTIC COMPLEXITY FOR MULTINOMIAL DATA

2.1 INTRODUCTION TO MDL

Let us consider a data set (or matrix) $\mathbf{x}^N = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ of N outcomes (vectors), where each outcome $\mathbf{x}_j$ is an element of the set $\mathcal{X}$. The set $\mathcal{X}$ consists of all the vectors of the form $(a_1, \ldots, a_m)$, where each variable (or attribute) a_i takes on values $v \in \{1, \ldots, n_i\}$. Furthermore, we assume that our data is multinomially distributed.

We now consider the case with a parametric family of *probabilistic candidate models* (or codes) $\mathcal{M} = \{f(\mathbf{x}|\theta) \mid \theta \in \Gamma\}$, where Γ is an open bounded region of $\mathbf{R}^k$ and k is a positive integer. The basic principle behind *Minimum Description Length (MDL)* modeling is to find a code that minimizes the code length over all data sequences which can be well modeled by $\mathcal{M}$. Here a data sequence being "well-modeled by $\mathcal{M}$" means that there is a model θ in $\mathcal{M}$ which gives a good fit to the data. In other words, if we let $\hat{\theta}(\mathbf{x}^N)$ denote the maximum likelihood estimator (MLE) of the data $\mathbf{x}^N$, then $\mathbf{x}^N$ is well modeled by $\mathcal{M}$ means that $f(\mathbf{x}^N|\hat{\theta}(\mathbf{x}^N))$ is high. The *stochastic complexity* of a data sequence $\mathbf{x}^N$, relative to a family of models $\mathcal{M}$, is the code length of $\mathbf{x}^N$ when it is encoded using the most efficient code obtainable with the help of the family $\mathcal{M}$.

In the above, stochastic complexity was defined only in an implicit manner — as discussed in (Grünwald, Kontkanen, Myllymäki, Silander, & Tirri, 1998), there exist several alternative ways for defining the stochastic complexity measure and the MDL principle explicitly. In (Rissanen, 1996) Rissanen shows how the two-part code MDL presented in (Rissanen, 1978) can be refined to a much more efficient coding scheme. This scheme is based on a notion of *normalized maximum likelihood (NML)*, proposed for finite alphabets in (Shtarkov, 1987). The definition of NML is

$$P_{NML}(\mathbf{x}^N \mid \mathcal{M}) = \frac{P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M})}{\sum_{\mathbf{y}^N} P(\mathbf{y}^N \mid \hat{\theta}(\mathbf{y}^N), \mathcal{M})}, \quad (1)$$

where the sum goes over all the possible data matrices of length N. For discussions on the theoretical motivations behind this criterion, see e.g. (Rissanen, 1996; Merhav & Feder, 1998; Barron et al., 1998; Grünwald, 1998; Rissanen, 1999; Xie & Barron, 2000; Rissanen, 2001).

Definition (1) is intuitively very appealing: every data matrix is coded using its own maximum likelihood (i.e. best fit) model, and then a penalty for the complexity of the model class $\mathcal{M}$ is added to normalize the distribution. This penalty, i.e., the denominator of (1), is called the *regret*. Note that usually the regret is defined as a logarithm of the

denominator. In this paper, however, we mostly use the language of probability theory rather than information theory and thus the definition without the logarithm is more natural.

2.2 COMPUTING THE NML: ONE-DIMENSIONAL CASE

We now turn to the question of how to compute the NML criterion (1), given a data matrix $\mathbf{x}^N$ and a model class $\mathcal{M}_1$. Let us first consider a case with only one multinomial variable with K values. The maximum likelihood term is easy and efficient to compute:

$$P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M}_1) = \prod_{j=1}^{N} P(\mathbf{x}_j \mid \hat{\theta}(\mathbf{x}^N)) = \prod_{v=1}^{K} \hat{\theta}_v^{h_v} = \prod_{v=1}^{K} \left(\frac{h_v}{N}\right)^{h_v}, \quad (2)$$

where $\hat{\theta}_v$ is the probability of value v, and $(h_1, \ldots, h_K)$ are the *sufficient statistics* of $\mathbf{x}^N$, which in the case of multinomial data are simply the frequencies of the values $\{1, \ldots, K\}$ in $\mathbf{x}^N$.

At first sight it may seem that the time complexity of computing the regret, i.e., the denominator in (1), grows exponentially with the size of the data, since the summing goes over K^N terms. However, it turns out that for reasonable small values of K it is possible to compute (1) efficiently. Since the maximum likelihood (2) only depends on the sufficient statistics h_v, the regret can be written as

$$R^1_{K,N} \stackrel{\text{def.}}{=} \sum_{\mathbf{x}^N} P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M}_1) = \sum_{h_1+\cdots+h_K=N} \frac{N!}{h_1! \cdots h_K!} \prod_{v=1}^{K} \left(\frac{h_v}{N}\right)^{h_v}, \quad (3)$$

where in the last formula the summing goes over all the *compositions* of N into K parts, i.e., over all the possible ways to choose non-negative integers $h_1, \ldots, h_K$ so that they sum up to N. We use the notation $R^1_{K,N}$ to refer to this subsequently, i.e., the regret for one multinomial variable with K values and N data vectors. The time complexity of (3) is $\mathcal{O}(N^{K-1})$, which is easy to see. For example, take case $K = 3$. The regret can be computed in $\mathcal{O}(N^2)$ time:

$$R^1_{3,N} = \sum_{h_1=0}^{N} \sum_{h_2=0}^{N-h_1} \frac{N!}{h_1! h_2! (N-h_1-h_2)!} \cdot \left(\frac{h_1}{N}\right)^{h_1} \left(\frac{h_2}{N}\right)^{h_2} \left(\frac{N-h_1-h_2}{N}\right)^{N-h_1-h_2}. \quad (4)$$

2.3 COMPUTING THE NML : THE RECURSIVE FORMULA

It turns out that the exact regret for a single multinomial variable can also be computed with a computationally very efficient combinatoric recursive formula. Consider $R^1_{K,N}$ as before. Using standard combinatorics we get the following recursion:

$$R^1_{K,N} = \sum_{h_1+h_2=N} \frac{N!}{h_1! h_2!} \left(\frac{h_1}{N}\right)^{h_1} \left(\frac{h_2}{N}\right)^{h_2} \cdot R^1_{k1,h1} R^1_{k2,h2}, \quad (5)$$

where $k_1 + k_2 = K$.

This formula allows us to compute the exact NML very efficiently by applying a common doubling trick from combinatorics. Firstly, one computes the tables of $R^1_{2^m,n}$ for $m = 1, \ldots, \lfloor \log K \rfloor$ and $n = 1, \ldots, N$. Secondly, $R^1_{K,N}$ can be built up from these tables. For example, take the case $R^1_{26,N}$. First calculate $R^1_{K,n}$ for $K \in \{2, 4, 8, 16\}$ and $n = 1, \ldots, N$. Then apply (5) to calculate the tables of $R^1_{10,n}$ from $R^1_{2,n}$ and $R^1_{8,n}$. Finally, $R^1_{26,N}$ can be computed from the tables of $R^1_{16,n}$ and $R^1_{10,n}$. It is now easy to see that the time complexity of computing (5) is $\mathcal{O}(N^2 \log K)$.

2.4 COMPUTING THE NML : MULTI-DIMENSIONAL CASE

The one-dimensional case discussed in the previous sections is not adequate for many real-world situations, where data is typically multi-dimensional. Let us assume that we have m variables. The number of possible data vectors is $\prod_{i=1}^{m} n_i$. It is clear that even the methods presented in the previous sections do not make the NML computation efficient in the multi-dimensional case. We are forced to make some independence assumptions. In this article, we assume the existence of a special variable c (which can be chosen to be one of the variables in our data matrix or it can be latent), and that given the value of c, the variables $(a_1, \ldots, a_m)$ are independent. That is, denoting the model class resulting from this assumption by $\mathcal{M}_T$,

$$P(c, a_1, \ldots, a_m \mid \theta, \mathcal{M}_T) = P(c \mid \theta, \mathcal{M}_T) \prod_{i=1}^{m} P(a_i \mid c, \theta, \mathcal{M}_T). \quad (6)$$

Although simple, this model class has been very successful in practice in mixture modeling (Kontkanen, Myllymäki, & Tirri, 1996), cluster analysis, case-based reasoning (Kontkanen, Myllymäki, Silander, & Tirri, 1998), Naive Bayes classification (Grünwald et al., 1998; Kontkanen, Myllymäki, Silander, Tirri, & Grünwald, 2000) and data visualization (Kontkanen, Lahtinen, Myllymäki, Silander, & Tirri, 2000).

We now show how to compute NML for $\mathcal{M}_T$. Assuming c has K values and using (3), Equation (1) becomes

$$P_{NML}(\mathbf{x}^N|\mathcal{M}_T) = \frac{\prod_{k=1}^{K}\left(\frac{h_k}{N}\right)^{h_k}\prod_{i=1}^{m}\prod_{k=1}^{K}\prod_{v=1}^{n_i}\left(\frac{f_{ikv}}{h_k}\right)^{f_{ikv}}}{R^m_{\mathcal{M}_T}}, \quad (7)$$

where h_k is the number of times c has value k in $\mathbf{x}^N$, f_{ikv} is the number of times a_i has value v when $c = k$, and $R^m_{\mathcal{M}_T}$ is the regret:

$$R^m_{\mathcal{M}_T} = \sum_{h_1+\cdots+h_K=N} \sum_{f_{111}+\cdots+f_{11n_1}=h_1} \cdots \sum_{f_{1K1}+\cdots+f_{1Kn_1}=h_K} \cdots \sum_{f_{m11}+\cdots+f_{m1n_m}=h_1} \cdots \sum_{f_{mK1}+\cdots+f_{mKn_m}=h_K} \frac{N!}{h_1!\cdots h_K!}\prod_{k=1}^{K}\left(\frac{h_k}{N}\right)^{h_k} \cdot \prod_{i=1}^{m}\prod_{k=1}^{K}\frac{h_k!}{f_{ik1}!\cdots f_{ikn_i}!}\prod_{v=1}^{n_i}\left(\frac{f_{ikv}}{h_k}\right)^{f_{ikv}}. \quad (8)$$

The trick to make (8) more efficient is to note that we can move all the terms under their respective summation signs, and replace the inner term with the one-dimensional case, which gives

$$R^m_{\mathcal{M}_T} = \sum_{h_1+\cdots+h_K=N} \frac{N!}{h_1!\cdots h_K!}\prod_{k=1}^{K}\left(\frac{h_k}{N}\right)^{h_k} \cdot \prod_{i=1}^{m}\prod_{k=1}^{K} R^1_{h_k,n_i}. \quad (9)$$

This depends only linearly on the number of variables m making it possible to compute (7) for cases with lots of variables provided that the number of value counts are reasonably small. On the other hand, formula (9) is clearly exponential with respect to K. This makes it infeasible for cases like cluster analysis, where typically K can be very big.

It turns out that the recursive formula (5) can also be generalized to the multi-dimensional case. There are, however, cases where even this recursive generalization is too inefficient. One important example is stochastic optimization problems, where typically one must evaluate the cost function thousands or even hundreds of thousands of times. It is clear that for these cases efficient approximations are needed. This will be the subject of the next section.

3 STOCHASTIC COMPLEXITY APPROXIMATIONS

In the previous section we discussed how the NML can be computed efficiently for both one- and multi-dimensional cases. However, we usually had to assume that the variables in our domain do not have too many values. Although the recursive formula (5) is only logarithmic with respect to the number of values, it is still quadratically dependent on the number of data vectors. Therefore, it is necessary to develop approximations to the NML. In this section, we are going to present three such approximations, two of which are well-known (BIC, Rissanen's asymptotic expansion) and a new one based on analytic combinatorics. For each approximation, we instantiate them for both the single multinomial case and the multivariate model class $\mathcal{M}_T$ defined by Equation (6). Furthermore, since we are able to compute the exact NML for these interesting and important cases, it is possible for the first time assess how accurate these approximations really are. This will be the topic of Section 4.

3.1 BAYESIAN INFORMATION CRITERION

The *Bayesian information criterion (BIC)* (Schwarz, 1978; Kass & Raftery, 1994), also known as the Schwarz criterion, is the simplest of the three approximations. For the single multinomial variable case, we get

$$-\log P_{BIC}(\mathbf{x}^N|\mathcal{M}_1) = -\log P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N)) + \frac{K-1}{2}\log(N), \quad (10)$$

where K is the number of values of the multinomial variable. As the name implies, the BIC has a Bayesian interpretation, but it can also be given a formulation in the MDL setting, as showed in (Rissanen, 1989).

In the multi-dimensional case, we easily get

$$-\log P_{BIC}(\mathbf{x}^N|\mathcal{M}_T) = -\log P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N)) + \frac{(K-1)+K\cdot\sum_{i=1}^{m}(n_i-1)}{2}\cdot\log(N). \quad (11)$$

As can be seen, the BIC approximation is very quick to compute and also easy to generalize to more complex model classes. However, it is known that BIC typically favors too simple model classes.

3.2 RISSANEN'S ASYMPTOTIC EXPANSION

As proved in (Rissanen, 1996), for model classes that satisfy certain regularity conditions, an asymptotic expansion can be derived. The most important condition is that the Central Limit Theorem should hold for the maximum likelihood estimators for all the elements in the model class.

The precise regularity conditions can be found in (Rissanen, 1996). The expansion is as follows:

$$-\log P_{RIS}(\mathbf{x}^N|\mathcal{M}) = -\log P(\mathbf{x}^N|\hat{\theta}(\mathbf{x}^N)) + \frac{k}{2}\log\frac{N}{2\,\pi} + \log\int\sqrt{|I(\theta)|}d\theta + o\,(1)\,, \quad (12)$$

where the integral goes over all the possible parameter vectors $\theta \in \mathcal{M}$, and $I(\theta)$ is the (expected) Fisher information matrix. The first term is the familiar negative logarithm of maximum likelihood. The second term measures the complexity that is due to the number of parameters in the model. Finally, the last term measures the complexity that comes from the local geometrical properties of the model space. For a more precise discussion, see (Grünwald, 1998).

Rissanen's asymptotic expansion for a single multinomial variable is discussed in (Rissanen, 1996), and with our notation it is given by

$$-\log P_{RIS}(\mathbf{x}^N|\mathcal{M}_1) = -\log P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N)) + \frac{K-1}{2}\log\left(\frac{N}{2\,\pi}\right) + \log\left(\frac{\pi^{K/2}}{\Gamma\left(\frac{K}{2}\right)}\right) + o\,(1)\,, \quad (13)$$

where $\Gamma(\cdot)$ is the Euler gamma function.

For the multi-dimensional case, we have earlier (Kontkanen et al., 2000) derived the square root of the determinant of the Fisher information for model class $\mathcal{M}_T$:

$$\sqrt{|I(\theta)|} = \prod_{k=1}^{K}\alpha_k^{\frac{1}{2}\left(\sum_{i=1}^{m}(n_i-1)-1\right)}\prod_{i=1}^{m}\prod_{k=1}^{K}\prod_{v=1}^{n_i}\theta_{ikv}^{-\frac{1}{2}}, \quad (14)$$

where $\alpha_k = P(c=k)$ and $\theta_{ikv} = P(a_i = v|c=k)$. To get (12), we need to integrate this expression over the parameters. Fortunately, this is relatively easy since this expression is a product of Dirichlet integrals, yielding

$$\begin{aligned}\int\sqrt{|I(\theta)|}d\theta &= \int\prod_{k=1}^{K}\alpha_k^{\frac{1}{2}\left(\sum_{i=1}^{m}(n_i-1)-1\right)}\cdot\prod_{i=1}^{m}\prod_{k=1}^{K}\prod_{v=1}^{n_i}\theta_{ikv}^{-\frac{1}{2}}d\theta \\ &= \frac{\prod_{k=1}^{K}\Gamma\left(\frac{1}{2}\left(\sum_{i=1}^{m}(n_i-1)+1\right)\right)}{\Gamma\left(\frac{K}{2}\left(\sum_{i=1}^{m}(n_i-1)+1\right)\right)}\cdot\prod_{i=1}^{m}\prod_{k=1}^{K}\frac{\pi^{n_i/2}}{\Gamma\left(\frac{n_i}{2}\right)}, \quad (15)\end{aligned}$$

and after simplifications we get

$$\begin{aligned}-\log P_{RIS}(\mathbf{x}^N|\mathcal{M}_T) &= -\log P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N)) \\ &+ \frac{(K-1)+K\sum_{i=1}^{m}(n_i-1)}{2}\log\left(\frac{N}{2\,\pi}\right) \\ &+ K\cdot\log\Gamma\left(\frac{1}{2}\left(\sum_{i=1}^{m}(n_i-1)+1\right)\right) \\ &- \log\Gamma\left(\frac{K}{2}\left(\sum_{i=1}^{m}(n_i-1)+1\right)\right) \\ &+ K\cdot\sum_{i=1}^{m}\left(\frac{n_i}{2}\log\pi - \log\Gamma\left(\frac{n_i}{2}\right)\right) + o\,(1)\,. \quad (16)\end{aligned}$$

Clearly, Rissanen's asymptotic expansion is efficient to compute, but for more complex model classes than our $\mathcal{M}_T$, the determinant of the Fisher information is no longer a product of Dirichlet integrals, which might cause technical problems.

3.3 SZPANKOWSKI APPROXIMATION

Theorem 8.32 in (Szpankowski, 2001) gives the redundancy rate for memoryless sources. The theorem is based on analytic combinatorics and generating functions, and can be used as a basis for a new NML approximation. Redundancy rate for memoryless sources is actually the regret for a single multinomial variable, and thus we have

$$\begin{aligned}-\log P_{SZP}(\mathbf{x}^N|\mathcal{M}_1) &= -\log P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N)) \\ &+ \frac{K-1}{2}\log\left(\frac{N}{2}\right) + \log\left(\frac{\sqrt{\pi}}{\Gamma\left(\frac{K}{2}\right)}\right) + \frac{\sqrt{2}\,\Gamma\left(\frac{K}{2}\right)}{3\sqrt{N}\,\Gamma\left(\frac{K}{2}-1/2\right)} \\ &+ \left(\frac{3+K(K-2)(2K+1)}{36} - \frac{\Gamma^2\left(\frac{K}{2}\right)K^2}{9\Gamma^2\left(\frac{K}{2}-1/2\right)}\right)\cdot\frac{1}{N} \\ &+ \mathcal{O}\left(\frac{1}{N^{3/2}}\right). \quad (17)\end{aligned}$$

For the multi-dimensional case we can use the factorized form (9) of the exact NML. Let $\hat{R}^1_{K,N}$ denote the regret approximation in (17) with N data vectors and K possible values. Now we can write

$$\begin{aligned}-\log P_{SZP}(\mathbf{x}^N|\mathcal{M}_T) &= -\log P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N)) \\ &+ \log\sum_{h_1+\cdots+h_K=N}\left(\frac{N!}{h_1!\cdots h_K!}\prod_{k=1}^{K}\left(\frac{h_k}{N}\right)^{h_k}\right. \\ &\left.\cdot\prod_{i=1}^{m}\prod_{k=1}^{K}\hat{R}^1_{n_i,h_k}\right) + \mathcal{O}\left(\frac{1}{N^{3/2}}\right). \quad (18)\end{aligned}$$

The time complexity of this approximation grows exponentially with K. However, we believe that similar approximation to (17) can be derived for model class $\mathcal{M}_T$ so that this exponentiality could be removed. This is a topic for future work.

4 EMPIRICAL RESULTS

As noted in the previous section, since we are able to compute the exact NML for model classes discussed in this paper, we have a unique opportunity to test how accurate the NML approximations really are. The first thing to notice is that since all three approximations presented contain the maximum likelihood term, we can ignore it in the comparisons and concentrate on the (log-)regret. Notice that since the regret is constant given the model class (i.e., it does not depend on observed data), we avoid the problem of trying to choose representative and unbiased data sets for the experiments.

We conducted two sets of experiments corresponding to the single multinomial case and the multivariate model class $\mathcal{M}_T$. In the following, we will use the following abbreviations for the approximations:

- BIC: Bayesian information criteria presented in Section 3.1.
- RIS: Rissanen's asymptotic expansion presented in Section 3.2.
- SZP: Szpankowski-based approximation presented in Section 3.3.

We start with the one-dimensional case. Figures 1, 2 and 3 show the differences between the three approximations and the exact log-regret as a function of the data size N with a different K, i.e., with a different number of values for the single variable. Cases with $K = 2$, $K = 4$ and $K = 9$ are shown.

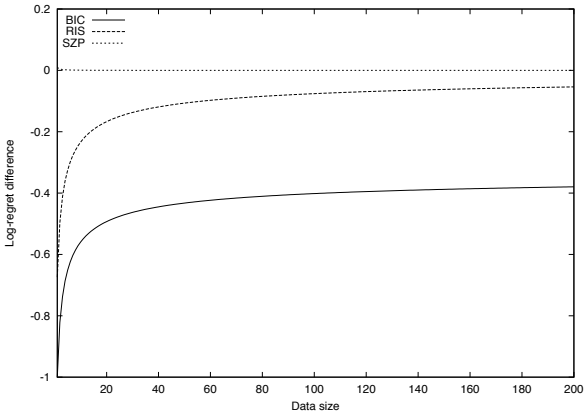

Figure 1: NML approximation results with a single multinomial variable having 2 values.

From these figures we see that the SZP approximation is clearly the best of the three. Furthermore, it is remarkably accurate: just after a few vectors the error is practically zero. The second best approximation is RIS, which takes about 100 data vectors or so to converge to a level near zero.

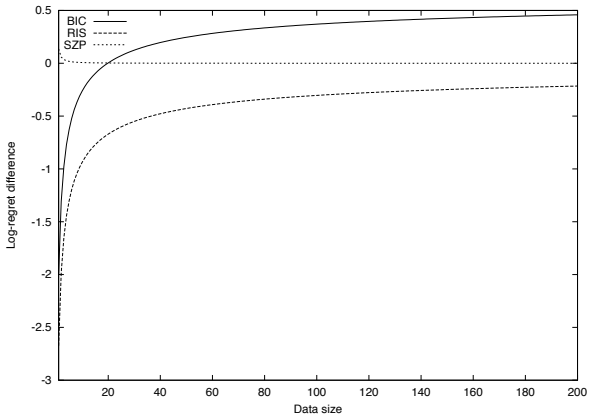

Figure 2: NML approximation results with a single multinomial variable having 4 values.

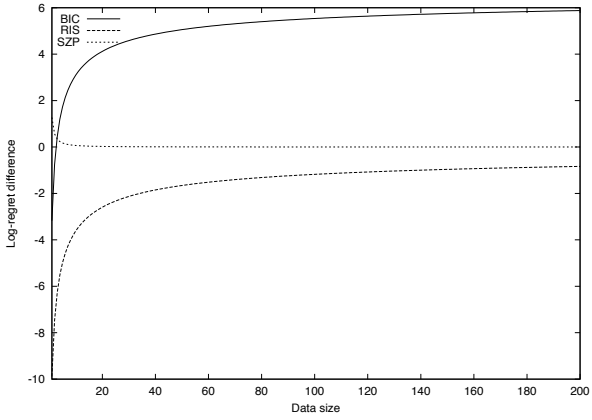

Figure 3: NML approximation results with a single multinomial variable having 9 values.

However, unlike SZP approximation, the convergence of RIS seems to get slower with increasing K. From figures 2 and 3 we see that when the test setting becomes more complex (with $K = 4$ and $K = 9$), BIC starts to overestimate the regret, and thus favors too simple models.

For the multidimensional case we tested with several values for the number of variables m. The results were very similar, so we show here only the case with 30 variables and 2 or 4 values. The special (clustering) variable c was taken to be binary in all tests. The results are shown in Figures 4 and 5.

From the results we can conclude that the SZP approximation is the best and prominently accurate approximation also in the multivariate case. Furthermore, it converged only after few data vectors also in this more complex setting. Rissanen's asymptotic expansion works still reasonably well, but the converge is slower than in the single multinomial case. The BIC approximation overestimates the regret in both cases, and becomes very inaccurate in more complex cases (as can be seen in Figure 5).

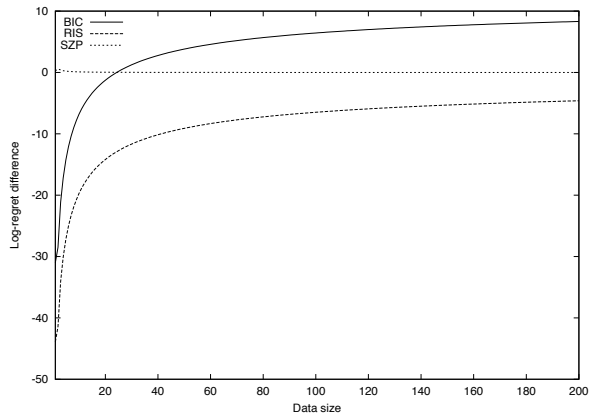

Figure 4: NML approximation results with 30 multinomial variables having 2 values.

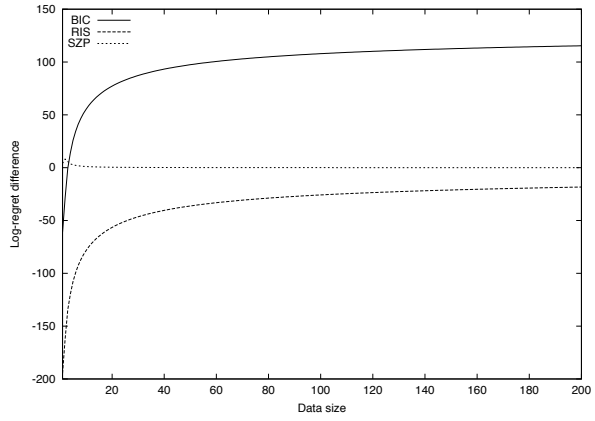

Figure 5: NML approximation results with 30 multinomial variables having 4 values.

5 CONCLUSION AND FUTURE WORK

In this article we have investigated how to compute the stochastic complexity both exactly and approximatively in an attempt to widen the application potential of the MDL principle. We showed that in the case of discrete data the exact form of SC can be computed for several important cases. Particularly interesting was the multi-dimensional model class case, which opens up several application possibilities for the MDL in problems like data clustering.

In addition to exact computation methods, we presented and instantiated three stochastic complexity approximations, and compared their accuracy. The most interesting and important observation was that the new approximation based on analytic combinatorics was significantly better than the older ones. It was also shown to be accurate already with very small sample sizes. Furthermore, the accuracy did not seem to get worse even for the more complex cases. This gives a clear indication that this approximation will also be useful for the cases where exact SC is not efficiently computable.

In the future, on the theoretical side, our goal is to extend the SZP approximation to more complex cases like general graphical models. Secondly, we will research supervised versions of SC, designed for supervised prediction tasks such as classification. On the application side, we have already conducted preliminary tests with MDL clustering by using proprietary real-world industrial data. The preliminary results are very encouraging: according to domain experts we have consulted, the clusterings found with MDL are much better than the ones found with traditional approaches. It is likely that the methods presented here can be used in several other application areas as well with similar success.

Acknowledgments

This research has been supported by the Academy of Finland.

References

Barron, A., Rissanen, J., & Yu, B. (1998). The minimum description principle in coding and modeling. *IEEE Transactions on Information Theory*, *44*(6), 2743–2760.

Bernardo, J. (1997). Noninformative priors do not exist. *J. Statist. Planning and Inference*, *65*, 159–189.

Bernardo, J., & Smith, A. (1994). *Bayesian theory*. John Wiley.

Bleistein, N., & Handelsman, R. (1975). *Asymptotic expansions of integrals*. Holt, Rinehart and Winston.

Chickering, D., & Heckerman, D. (1997). Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, *29*(2/3), 181–212.

Clarke, B., & Barron, A. (1990). Information-theoretic asymptotics of Bayes methods. *IEEE Transactions on Information Theory*, *36*(3), 453–471.

Dom, B. (1995). *MDL estimation with small sample sizes including an application to the problem of segmenting binary strings using Bernoulli models* (Tech. Rep. No. RJ 9997 (89085)). IBM Research Division, Almaden Research Center.

Dom, B. (1996). *MDL estimation for small sample sizes and its application to linear regression* (Tech. Rep. No. RJ 10030 (90526)). IBM Research Division, Almaden Research Center.

Grünwald, P. (1998). *The minimum description length principle and reasoning under uncertainty.*

Ph.D. Thesis, CWI, ILLC Dissertation Series 1998-03.

Grünwald, P., Kontkanen, P., Myllymäki, P., Silander, T., & Tirri, H. (1998). Minimum encoding approaches for predictive modeling. In G. Cooper & S. Moral (Eds.), *Proceedings of the 14th international conference on uncertainty in artificial intelligence (UAI'98)* (pp. 183–192). Madison, WI: Morgan Kaufmann Publishers, San Francisco, CA.

Kass, R., & Raftery, A. (1994). *Bayes factors* (Tech. Rep. No. 254). Department of Statistics, University of Washington.

Kontkanen, P., Lahtinen, J., Myllymäki, P., Silander, T., & Tirri, H. (2000). Supervised model-based visualization of high-dimensional data. *Intelligent Data Analysis*, *4*, 213–227.

Kontkanen, P., Myllymäki, P., Silander, T., & Tirri, H. (1998). On Bayesian case matching. In B. Smyth & P. Cunningham (Eds.), *Advances in case-based reasoning, proceedings of the 4th european workshop (EWCBR-98)* (Vol. 1488, pp. 13–24). Springer-Verlag.

Kontkanen, P., Myllymäki, P., Silander, T., & Tirri, H. (1999). On the accuracy of stochastic complexity approximations. In A. Gammerman (Ed.), *Causal models and intelligent data management*. Springer-Verlag.

Kontkanen, P., Myllymäki, P., Silander, T., Tirri, H., & Grünwald, P. (2000). On predictive distributions and Bayesian networks. *Statistics and Computing*, *10*, 39–54.

Kontkanen, P., Myllymäki, P., & Tirri, H. (1996). *Constructing Bayesian finite mixture models by the EM algorithm* (Tech. Rep. No. NC-TR-97-003). ESPRIT Working Group on Neural and Computational Learning (NeuroCOLT).

Merhav, N., & Feder, M. (1998). Universal prediction. *IEEE Transactions on Information Theory*, *44*(6), 2124–2147.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 445-471.

Rissanen, J. (1987). Stochastic complexity. *Journal of the Royal Statistical Society*, *49*(3), 223–239 and 252–265.

Rissanen, J. (1989). *Stochastic complexity in statistical inquiry*. New Jersey: World Scientific Publishing Company.

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

Rissanen, J. (1999). Hypothesis selection and testing by the MDL principle. *Computer Journal*, *42*(4), 260–269.

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

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

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

Szpankowski, W. (2001). *Average case analysis of algorithms on sequences*. John Wiley & Sons.

Tabus, I., Rissanen, J., & Astola, J. (2002). *Classification and feature gene selection using the normalized maximum likelihood model for discrete regression*. (Unpublished manuscript, Institute of Signal Processing, Tampere University of Technology, Finland)

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

Paper II

P. Kontkanen, P. Myllymäki

A Fast Normalized Maximum Likelihood Algorithm for Multinomial Data

Pp. 1613-1616 in Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05), 2005.

A Fast Normalized Maximum Likelihood Algorithm for Multinomial Data

Petri Kontkanen, Petri Myllymäki
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
University of Helsinki & Helsinki University of Technology
P.O. Box 9800, FIN-02015 HUT, Finland.
{Firstname}.{Lastname}@hiit.fi

Abstract

Stochastic complexity of a data set is defined as the shortest possible code length for the data obtainable by using some fixed set of models. This measure is of great theoretical and practical importance as a tool for tasks such as model selection or data clustering. In the case of multinomial data, computing the modern version of stochastic complexity, defined as the Normalized Maximum Likelihood (NML) criterion, requires computing a sum with an exponential number of terms. Furthermore, in order to apply NML in practice, one often needs to compute a whole table of these exponential sums. In our previous work, we were able to compute this table by a recursive algorithm. The purpose of this paper is to significantly improve the time complexity of this algorithm. The techniques used here are based on the discrete Fourier transform and the convolution theorem.

1 Introduction

The *Minimum Description Length (MDL)* principle developed by Rissanen [Rissanen, 1978; 1987; 1996] offers a well-founded theoretical formalization of statistical modeling. The main idea of this principle is to represent a set of models (model class) by a single model imitating the behaviour of any model in the class. Such representative models are called *universal*. The universal model itself does not have to belong to the model class as often is the case.

From a computer science viewpoint, the fundamental idea of the MDL principle is *compression of data*. That is, given some sample data, the task is to find a description or *code* of the data such that this description uses less symbols than it takes to describe the data literally. Intuitively speaking, this approach can in principle be argued to produce the best possible model of the problem domain, since in order to be able to produce the most efficient coding of data, one must capture all the regularities present in the domain.

The MDL principle has gone through several evolutionary steps during the last two decades. For example, the early realization of the MDL principle, the two-part code MDL [Rissanen, 1978], takes the same form as the Bayesian BIC criterion [Schwarz, 1978], which has led some people to incorrectly believe that MDL and BIC are equivalent. The latest instantiation of the MDL is *not* directly related to BIC, but to the formalization described in [Rissanen, 1996]. Unlike Bayesian and many other approaches, the modern MDL principle does not assume that the chosen model class is correct. It even says that there is no such thing as a true model or model class, as acknowledged by many practitioners. The model class is only used as a technical device for constructing an efficient code. For discussions on the theoretical motivations behind the modern definition of the MDL see, e.g., [Rissanen, 1996; Merhav and Feder, 1998; Barron *et al*., 1998; Grünwald, 1998; Rissanen, 1999; Xie and Barron, 2000; Rissanen, 2001].

The most important notion of the MDL principle is the *Stochastic Complexity (SC)*, which is defined as the shortest description length of a given data relative to a model class $\mathcal{M}$. The modern definition of SC is based on the Normalized Maximum Likelihood (NML) code [Shtarkov, 1987]. Unfortunately, with multinomial data this code involves a sum over all the possible data matrices of certain length. Computing this sum, usually called the *regret*, is obviously exponential. Therefore, practical applications of the NML have been quite rare,

In our previous work [Kontkanen *et al*., 2003; 2005], we presented a polynomial time (quadratic) method to compute the regret. In this paper we improve our previous results and show how mathematical techniques such as discrete Fourier transform and convolution can be used in regret computation. The idea of applying these techniques for computing a single regret term was first suggested in [Koivisto, 2004], but as discussed in [Kontkanen *et al*., 2005], in order to apply NML to practical tasks such as clustering, a whole table of regret terms is needed. We will present here an efficient algorithm for this specific task. For a more detailed discussion of this work, see [Kontkanen and Myllymäki, 2005].

2 NML for Multinomial Data

The most important notion of the MDL is the *Stochastic Complexity (SC)*. Intuitively, stochastic complexity is defined as the shortest description length of a given data relative to a model class. To formalize things, let us start with a definition of a model class. Consider a set $\Theta \in \mathbb{R}^d$, 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

$$\mathcal{M} = \{P(\cdot \mid \theta) : \theta \in \Theta\}. \quad (1)$$

Consider now a discrete data set (or matrix) $\mathbf{x}^N = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ of N outcomes, where each outcome $\mathbf{x}_j$ is an element of the set $\mathcal{X}$ consisting of all the vectors of the form $(a_1, \ldots, a_m)$, where each variable (or attribute) a_i takes on values $v \in \{1, \ldots, n_i\}$. Given a model class $\mathcal{M}$, the *Normalized Maximum Likelihood (NML)* distribution [Shtarkov, 1987] is defined as

$$P_{NML}(\mathbf{x}^N \mid \mathcal{M}) = \frac{P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M})}{\mathcal{R}^N_{\mathcal{M}}}, \quad (2)$$

where $\hat{\theta}(\mathbf{x}^N)$ denotes the *maximum likelihood* estimate of data $\mathbf{x}^N$, and $\mathcal{R}^N_{\mathcal{M}}$ is given by

$$\mathcal{R}^N_{\mathcal{M}} = \sum_{\mathbf{x}^N} P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M}), \quad (3)$$

and the sum goes over all the possible data matrices of size N. The term $\mathcal{R}^N_{\mathcal{M}}$ is called the *regret*. The definition (2) is intuitively very appealing: every data matrix is modeled using its own maximum likelihood (i.e., best fit) model, and then a penalty for the complexity of the model class $\mathcal{M}$ is added to normalize the distribution.

The stochastic complexity of a data set $\mathbf{x}^N$ with respect to a model class $\mathcal{M}$ can now be defined as the negative logarithm of (2), i.e.,

$$SC(\mathbf{x}^n \mid \mathcal{M}) = -\log \frac{P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M})}{\mathcal{R}^N_{\mathcal{M}}} \quad (4)$$

$$= -\log P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M}) + \log \mathcal{R}^N_{\mathcal{M}}. \quad (5)$$

As in [Kontkanen *et al*., 2005], in the sequel we focus on a multi-dimensional model class suitable for cluster analysis. The selected model class has also been successfully applied to mixture modeling [Kontkanen *et al*., 1996], case-based reasoning [Kontkanen *et al*., 1998], Naive Bayes classification [Grünwald *et al*., 1998; Kontkanen *et al*., 2000b] and data visualization [Kontkanen *et al*., 2000a].

Let us assume that we have m variables, $(a_1, \ldots, a_m)$, and we also assume the existence of a special variable c (which can be chosen to be one of the variables in our data or it can be latent). Furthermore, given the value of c, the variables $(a_1, \ldots, a_m)$ are assumed to be independent. The resulting model class is denoted by $\mathcal{M}_T$. Suppose the special variable c has K values and each a_i has n_i values. The NML distribution for the model class $\mathcal{M}_T$ is now

$$P_{NML}(\mathbf{x}^N \mid \mathcal{M}_T) = \left[\prod_{k=1}^{K} \left(\frac{h_k}{N}\right)^{h_k} \prod_{i=1}^{m} \prod_{k=1}^{K} \prod_{v=1}^{n_i} \left(\frac{f_{ikv}}{h_k}\right)^{f_{ikv}}\right] \cdot \frac{1}{\mathcal{R}^N_{\mathcal{M}_T,K}}, \quad (6)$$

where h_k is the number of times c has value k in $\mathbf{x}^N$, f_{ikv} is the number of times a_i has value v when $c = k$, and $\mathcal{R}^N_{\mathcal{M}_T,K}$ is the regret term. In [Kontkanen *et al*., 2005] it was proven that an efficient way to compute the regret term is via the following recursive formula:

$$\mathcal{R}^N_{\mathcal{M}_T,K} = \sum_{r=0}^{N} \frac{N!}{r!(N-r)!} \left(\frac{r}{N}\right)^r \left(\frac{N-r}{N}\right)^{N-r} \cdot \mathcal{R}^r_{\mathcal{M}_T,k_1} \cdot \mathcal{R}^{N-r}_{\mathcal{M}_T,k_2}, \quad (7)$$

where $k_1 + k_2 = K$.

As discussed in [Kontkanen *et al*., 2005], in order to apply NML to the clustering problem, we need to compute a whole table of regret terms. This table consists of the terms $\mathcal{R}^n_{\mathcal{M}_T,k}$ for $n = 0, \ldots, N$ and $k = 1, \ldots, K$, where K is the maximum number of clusters.

The procedure of computing the regret table starts by filling the first column, i.e., the case $k = 1$, which is trivial (see [Kontkanen *et al*., 2005]). To compute the column k, for $k = 2, \ldots, K$, the recursive formula (7) can be used by choosing $k_1 = k - 1$, $k_2 = 1$. The time complexity of filling the whole table is $\mathcal{O}\left(K \cdot N^2\right)$. For more details, see [Kontkanen *et al*., 2005; Kontkanen and Myllymäki, 2005].

In practice, the quadratic dependency on the size of data limits the applicability of NML to small or moderate size data sets. In the next section, we will present a novel, significantly more efficient method for computing the regret table.

3 The Fast NML Algorithm

In this section we will derive a very efficient algorithm for the regret table computation. The new method is based on the Fast Fourier Transform algorithm. As mentioned in the previous section, the calculation of the first column of the regret table is trivial. Therefore, we only need to consider the case of calculating the column k given the first $k-1$ columns. Let us define two sequences $\mathbf{a}$ and $\mathbf{b}$ by

$$a_n = \frac{n^n}{n!}\mathcal{R}^n_{\mathcal{M}_T,k-1}, \quad b_n = \frac{n^n}{n!}\mathcal{R}^n_{\mathcal{M}_T,1}, \quad (8)$$

for $n = 0, \ldots, N$. Evaluating the convolution of $\mathbf{a}$ and $\mathbf{b}$ gives

$$(\mathbf{a} * \mathbf{b})_n = \sum_{h=0}^{n} \frac{h^h}{h!}\mathcal{R}^h_{\mathcal{M}_T,k-1} \frac{(n-h)^{n-h}}{(n-h)!}\mathcal{R}^{n-h}_{\mathcal{M}_T,1} \quad (9)$$

$$= \frac{n^n}{n!} \sum_{h=0}^{n} \frac{n!}{h!(n-h)!} \left(\frac{h}{n}\right)^h \left(\frac{n-h}{n}\right)^{n-h} \cdot \mathcal{R}^h_{\mathcal{M}_T,k-1}\mathcal{R}^{n-h}_{\mathcal{M}_T,1} \quad (10)$$

$$= \frac{n^n}{n!}\mathcal{R}^n_{\mathcal{M}_T,k}, \quad (11)$$

where the last equality follows from the recursion formula (7). This derivation shows that the column k can be computed by first evaluating the convolution (11), and then multiplying each term by $n!/n^n$.

The standard *convolution theorem* states that convolutions can be evaluated via the (discrete) Fourier transform, which in turn can be computed efficiently with the Fast Fourier Transform algorithm (see [Kontkanen and Myllymäki, 2005] for details). It follows that the time complexity of computing the

whole regret table drops to $\mathcal{O}(N \log N \cdot K)$. This is a major improvement over $\mathcal{O}\left(N^2 \cdot K\right)$ obtained by the recursion method of Section 2.

4 Conclusion And Future Work

The main result of this paper was a derivation of a novel algorithm for the regret table computation. The theoretical time complexity of this algorithm allows practical applications of NML in domains with very large datasets. With the earlier quadratic-time algorithms, this was not possible.

In the future, we plan to conduct an extensive set of empirical tests to see how well the theoretical advantage of the new algorithm transfers to practice. On the theoretical side, our goal is to extend the regret table computation to more complex cases like general graphical models. We will also research supervised versions of the stochastic complexity, designed for supervised prediction tasks such as classification.

Acknowledgements

This work was supported in part by the Academy of Finland under the projects Minos and Civi and by the National Technology Agency under the PMMA project. 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

[Barron *et al*., 1998] A. Barron, J. Rissanen, and B. Yu. The minimum description principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, October 1998.

[Grünwald *et al*., 1998] P. Grünwald, P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. Minimum encoding approaches for predictive modeling. In G. Cooper and S. Moral, editors, *Proceedings of the 14th International Conference on Uncertainty in Artificial Intelligence (UAI'98)*, pages 183–192, Madison, WI, July 1998. Morgan Kaufmann Publishers, San Francisco, CA.

[Grünwald, 1998] P. Grünwald. *The Minimum Description Length Principle and Reasoning under Uncertainty*. PhD thesis, CWI, ILLC Dissertation Series 1998-03, 1998.

[Koivisto, 2004] 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.

[Kontkanen and Myllymäki, 2005] P. Kontkanen and P. Myllymäki. Computing the regret table for multinomial data. Technical Report 2005-1, Helsinki Institute for Information Technology (HIIT), 2005.

[Kontkanen *et al*., 1996] P. Kontkanen, P. Myllymäki, and H. Tirri. Constructing Bayesian finite mixture models by the EM algorithm. Technical Report NC-TR-97-003, ESPRIT Working Group on Neural and Computational Learning (NeuroCOLT), 1996.

[Kontkanen *et al*., 1998] P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. On Bayesian case matching. In B. Smyth and P. Cunningham, editors, *Advances in Case-Based Reasoning, Proceedings of the 4th European Workshop (EWCBR-98)*, volume 1488 of *Lecture Notes in Artificial Intelligence*, pages 13–24. Springer-Verlag, 1998.

[Kontkanen *et al*., 2000a] P. Kontkanen, J. Lahtinen, P. Myllymäki, T. Silander, and H. Tirri. Supervised model-based visualization of high-dimensional data. *Intelligent Data Analysis*, 4:213–227, 2000.

[Kontkanen *et al*., 2000b] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald. On predictive distributions and Bayesian networks. *Statistics and Computing*, 10:39–54, 2000.

[Kontkanen *et al*., 2003] P. Kontkanen, W. Buntine, P. Myllymäki, J. Rissanen, and H. Tirri. Efficient computation of stochastic complexity. In C. Bishop and B. Frey, editors, *Proceedings of the Ninth International Conference on Artificial Intelligence and Statistics*, pages 233–238. Society for Artificial Intelligence and Statistics, 2003.

[Kontkanen *et al*., 2005] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri. An MDL framework for data clustering. In P. Grünwald, I.J. Myung, and M. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*. The MIT Press, 2005.

[Merhav and Feder, 1998] N. Merhav and M. Feder. Universal prediction. *IEEE Transactions on Information Theory*, 44(6):2124–2147, October 1998.

[Rissanen, 1978] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:445–471, 1978.

[Rissanen, 1987] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society*, 49(3):223–239 and 252–265, 1987.

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

[Rissanen, 1999] J. Rissanen. Hypothesis selection and testing by the MDL principle. *Computer Journal*, 42(4):260–269, 1999.

[Rissanen, 2001] 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, July 2001.

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

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

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

Paper III

P. Kontkanen and P. Myllymäki

A linear-time algorithm for computing the multinomial stochastic complexity

Information Processing Letters 103 (2007) 6 (September), 227-233.

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.

Count the frequencies $h_1, \ldots, h_K$ from the data $\mathbf{x}^n$
Compute the likelihood
$P\big(\mathbf{x}^n \mid \hat{\boldsymbol{\theta}}(\mathbf{x}^n, \mathcal{M}_K)\big) = \prod_{k=1}^{K} \big(\frac{h_k}{n}\big)^{h_k}$
Set $\mathcal{C}(1, n) = 1$
Compute
$\mathcal{C}(2, n) = \sum_{r_1+r_2=n} \frac{n!}{r_1! r_2!} \big(\frac{r_1}{n}\big)^{r_1} \big(\frac{r_2}{n}\big)^{r_2}$
for $k = 1$ to $K - 2$
  Compute
  $\mathcal{C}(k+2, n) = \mathcal{C}(k+1, n) + \frac{n}{k} \cdot \mathcal{C}(k, n)$
end for
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.

Paper IV

P. Kontkanen and P. Myllymäki

MDL Histogram Density Estimation

In Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007), Puerto Rico, March 2007.

MDL Histogram Density Estimation

Petri Kontkanen, Petri Myllymäki
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
University of Helsinki and Helsinki University of Technology
P.O.Box 68 (Department of Computer Science)
FIN-00014 University of Helsinki, Finland
{Firstname}.{Lastname}@hiit.fi

Abstract

We regard histogram density estimation as a model selection problem. Our approach is based on the information-theoretic minimum description length (MDL) principle, which can be applied for tasks such as data clustering, density estimation, image denoising and model selection in general. MDL-based model selection is formalized via the normalized maximum likelihood (NML) distribution, which has several desirable optimality properties. We show how this framework can be applied for learning generic, irregular (variable-width bin) histograms, and how to compute the NML model selection criterion efficiently. We also derive a dynamic programming algorithm for finding both the MDL-optimal bin count and the cut point locations in polynomial time. Finally, we demonstrate our approach via simulation tests.

1 INTRODUCTION

Density estimation is one of the central problems in statistical inference and machine learning. Given a sample of observations, the goal of *histogram density estimation* is to find a piecewise constant density that describes the data best according to some predetermined criterion. Although histograms are conceptually simple densities, they are very flexible and can model complex properties like multi-modality with a relatively small number of parameters. Furthermore, one does not need to assume any specific form for the underlying density function: given enough bins, a histogram estimator adapts to any kind of density.

Most existing methods for learning histogram densities assume that the bin widths are equal and concentrate only on finding the optimal bin count. These *regular* histograms are, however, often problematic. It has been argued (Rissanen, Speed, & Yu, 1992) that regular histograms are only good for describing roughly uniform data. If the data distribution is strongly non-uniform, the bin count must necessarily be high if one wants to capture the details of the high density portion of the data. This in turn means that an unnecessary large amount of bins is wasted in the low density region.

To avoid the problems of regular histograms one must allow the bins to be of variable width. For these *irregular* histograms, it is necessary to find the optimal set of *cut points* in addition to the number of bins, which naturally makes the learning problem essentially more difficult. For solving this problem, we regard the histogram density estimation as a model selection task, where the cut point sets are considered as models. In this framework, one must first choose a set of candidate cut points, from which the optimal model is searched for. The quality of each of the cut point sets is then measured by some model selection criterion.

Our approach is based on information theory, more specifically on the *Minimum description length* (MDL) principle developed in the series of papers (Rissanen, 1978, 1987, 1996). MDL is a well-founded, general framework for performing model 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 the least number of symbols, less than other codes and less 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.

Model selection with MDL is done by minimizing a

quantity called *the stochastic complexity*, which is the shortest description length of a given data relative to a given model class. The definition of the stochastic complexity is based on the *normalized maximum likelihood* (NML) distribution introduced in (Shtarkov, 1987; Rissanen, 1996). The NML distribution has several theoretical optimality properties, which make it a very attractive candidate for performing model selection. It was originally (Rissanen, 1996) formulated as a unique solution to the minimax problem presented in (Shtarkov, 1987), which implied that NML is the minimax optimal universal model. Later (Rissanen, 2001), it was shown that NML is also the solution to a related problem involving expected regret. See Section 2 and (Rissanen, 2001; Grünwald, 2006; Rissanen, 2005) 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 two examples. In (Kontkanen, Myllymäki, Buntine, Rissanen, & Tirri, 2006), NML was used for data clustering, and its performance was compared to alternative approaches like Bayesian statistics. The results showed that NML was especially impressive with small sample sizes. In (Roos, Myllymäki, & Tirri, 2005), NML was applied to wavelet denoising of computer images. Since the MDL principle in general can be interpreted as separating information from noise, this approach is very natural.

Unfortunately, in most practical applications of NML one must face severe computational problems, since the definition of the NML involves a normalizing integral or a sum, called the *parametric complexity*, which usually is difficult to compute. One of the contributions of this paper is to show how the parametric complexity can be computed efficiently in the histogram case, which makes it possible to use NML as a model selection criterion in practice.

There is obviously an exponential number of different cut point sets. Therefore, a brute-force search is not feasible. Another contribution of this paper is to show how the NML-optimal cut point locations can be found via dynamic programming in a polynomial (quadratic) time with respect to the size of the set containing the cut points considered in the optimization process.

The histogram density estimation is naturally a well-studied problem, but unfortunately almost all of the previous studies, e.g. (Birge & Rozenholc, 2002; Hall & Hannan, 1988; Yu & Speed, 1992), consider regular histograms only. Most similar to our work is (Rissanen et al., 1992), in which irregular histograms are learned with the Bayesian mixture criterion using a uniform prior. The same criterion is also used in (Hall & Hannan, 1988), but the histograms are equal-width only. Another similarity between our work and (Rissanen et al., 1992) is the dynamic programming optimization process, but since the optimality criterion is not the same, the process itself is quite different. It should be noted that these differences are significant as the Bayesian mixture criterion does not possess the optimality properties of NML mentioned above.

This paper is structured as follows. In Section 2 we discuss the basic properties of the MDL framework in general, and also shortly review the optimality properties of the NML distribution. Section 3 introduces the NML histogram density and also provides a solution to the related computational problem. The cut point optimization process based on dynamic programming is the topic of Section 4. Finally, in Section 5 our approach is demonstrated via simulation tests.

2 PROPERTIES OF MDL AND NML

The MDL principle has several desirable properties. Firstly, it automatically protects against overfitting when learning both the parameters and the structure (number of parameters) of the model. Secondly, there is no need to assume the existence of some underlying "true" model, which is not the case with several other statistical methods. The model is only used as a technical device for constructing an efficient code. MDL is also closely related to the 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.

MDL model 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 $\mathcal{M} = \{f(\cdot \mid \theta) : \theta \in \Theta\}$. Denote the maximum likelihood estimate of data $\mathbf{x}^n$ by $\hat{\theta}(\mathbf{x}^n)$, i.e.,

$$\hat{\theta}(\mathbf{x}^n) = \arg\max_{\theta \in \Theta} \{f(\mathbf{x}^n \mid \theta)\}. \tag{1}$$

The *normalized maximum likelihood (NML)* density (Shtarkov, 1987) is now defined as

$$f_{\mathrm{NML}}(\mathbf{x}^n \mid \mathcal{M}) = \frac{f(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), \mathcal{M})}{\mathcal{R}^n_{\mathcal{M}}}, \tag{2}$$

where the normalizing constant $\mathcal{R}^n_{\mathcal{M}}$ is given by

$$\mathcal{R}^n_{\mathcal{M}} = \int_{\mathbf{x}^n \in \mathcal{X}^n} f(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), \mathcal{M}) d\mathbf{x}^n, \quad (3)$$

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

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

$$SC(\mathbf{x}^n \mid \mathcal{M}) = -\log f_{\mathrm{NML}}(\mathbf{x}^n \mid \mathcal{M}) \quad (4)$$

$$= -\log f(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), \mathcal{M}) + \log \mathcal{R}^n_{\mathcal{M}}, \quad (5)$$

and the term $\log \mathcal{R}^n_{\mathcal{M}}$ is called the *parametric complexity* or *minimax regret*. The parametric complexity 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 (Balasubramanian, 2006) for more discussion on this topic.

The NML density (2) has several important theoretical optimality properties. The first one is that NML provides a unique solution to the minimax problem posed in (Shtarkov, 1987),

$$\min_{\hat{f}} \max_{\mathbf{x}^n} \log \frac{f(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), \mathcal{M})}{\hat{f}(\mathbf{x}^n \mid \mathcal{M})} = \log \mathcal{R}^n_{\mathcal{M}}, \quad (6)$$

This means that the NML density is the *minimax optimal universal model*. A related property of NML involving expected regret was proven in (Rissanen, 2001). This property states that NML also minimizes

$$\min_{\hat{f}} \max_{g} E_g \log \frac{f(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), \mathcal{M})}{\hat{f}(\mathbf{x}^n \mid \mathcal{M})} = \log \mathcal{R}^n_{\mathcal{M}}, \quad (7)$$

where the expectation is taken over $\mathbf{x}^n$ and g is the worst-case data generating density.

Having now discussed the MDL principle and the NML density in general, we return to the main topic of the paper. In the next section, we instantiate the NML density for the histograms and show how the parametric complexity can be computed efficiently in this case.

3 NML HISTOGRAM DENSITY

Consider a sample of n outcomes $\mathbf{x}^n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ on the interval $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$. Typically, $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ are defined as the minimum and maximum value in $\mathbf{x}^n$, respectively. Without any loss of generality, we assume that the data is sorted into increasing order. Furthermore, we assume that the data is recorded at a finite accuracy ϵ, which means that each $\mathbf{x}_j \in \mathbf{x}^n$ belongs to the set $\mathcal{X}$ defined by

$$\mathcal{X} = \{\mathbf{x}_{\min} + t\epsilon : t = 0, \ldots, \frac{\mathbf{x}_{\max} - \mathbf{x}_{\min}}{\epsilon}\}. \quad (8)$$

This assumption is made to simplify the mathematical formulation, and as can be seen later, the effect of the accuracy parameter ϵ on the stochastic complexity is a constant that can be ignored in the model selection process.

Let $C = (c_1, \ldots, c_{K-1})$ be an increasing sequence of points partitioning the range $[\mathbf{x}_{\min} - \epsilon/2, \mathbf{x}_{\max} + \epsilon/2]$ into the following K intervals (bins):

$$([\mathbf{x}_{\min} - \epsilon/2, c_1],]c_1, c_2], \ldots,]c_{K-1}, \mathbf{x}_{\max} + \epsilon/2]). \quad (9)$$

The points c_k are called the *cut points* of the histogram. Note that the original data range $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$ is extended by $\epsilon/2$ from both ends for technical reasons. It is natural to assume that there is only one cut point between two consecutive elements of $\mathcal{X}$, since placing two or more cut points would always produce unnecessary empty bins. For simplicity, we assume that the cut points belong to the set $\mathcal{C}$ defined by

$$\mathcal{C} = \{\mathbf{x}_{\min} + \epsilon/2 + t\epsilon : t = 0, \ldots, \frac{\mathbf{x}_{\max} - \mathbf{x}_{\min}}{\epsilon} - 1\}, \quad (10)$$

i.e., each $c_k \in \mathcal{C}$ is a midpoint of two consecutive values of $\mathcal{X}$.

Define $c_0 = \mathbf{x}_{\min} - \epsilon/2, c_K = \mathbf{x}_{\max} + \epsilon/2$ and let $L_k = c_k - c_{k-1}, k = 1, \ldots, K$ be the bin lengths. Given a parameter vector $\theta \in \Theta$,

$$\Theta = \{(\theta_1, \ldots, \theta_K) : \theta_k \geq 0, \theta_1 + \cdots + \theta_K = 1\}, \quad (11)$$

and a set (sequence) of cut points C, we now define the histogram density f_h by

$$f_h(x \mid \theta, C) = \frac{\epsilon \cdot \theta_k}{L_k}, \quad (12)$$

where $x \in]c_{k-1}, c_k]$. Note that (12) does not define a density in the purest sense, since $f_h(x \mid \theta, C)$ is actually the probability that x falls into the interval $]x - \epsilon/2, x + \epsilon/2]$. Given (12), the likelihood of the whole data sample $\mathbf{x}^n$ is easy to write. We have

$$f_h(\mathbf{x}^n \mid \theta, C) = \prod_{k=1}^{K} \left(\frac{\epsilon \cdot \theta_k}{L_k}\right)^{h_k}, \quad (13)$$

where h_k is the number of data points falling into bin k.

To instantiate the NML distribution (2) for the histogram density f_h, we need to find the maximum likelihood parameters $\hat{\theta}(x^n) = (\hat{\theta}_1, \ldots, \hat{\theta}_K)$ and an efficient way to compute the parametric complexity (3). It is well-known that the ML parameters are given by the relative frequencies $\hat{\theta}_k = h_k/n$, so that we have

$$f_h(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), C) = \prod_{k=1}^{K} \left(\frac{\epsilon \cdot h_k}{L_k \cdot n} \right)^{h_k}. \quad (14)$$

Denote now the parametric complexity of a K-bin histogram by $\log \mathcal{R}^n_{h_K}$. First thing to notice is that since the data is pre-discretized, the integral in (3) is replaced by a sum over the space $\mathcal{X}^n$. We have

$$\begin{aligned} \mathcal{R}^n_{h_K} &= \sum_{\mathbf{x}^n \in \mathcal{X}^n} \prod_{k=1}^{K} \left(\frac{\epsilon \cdot h_k}{L_k \cdot n} \right)^{h_k} && (15) \\ &= \sum_{h_1 + \cdots + h_K = n} \frac{n!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{L_k}{\epsilon} \right)^{h_k} \\ &\qquad \cdot \prod_{k=1}^{K} \left(\frac{\epsilon \cdot h_k}{L_k \cdot n} \right)^{h_k} && (16) \\ &= \sum_{h_1 + \cdots + h_K = n} \frac{n!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{n} \right)^{h_k}, && (17) \end{aligned}$$

where the term $(L_k/\epsilon)^{h_k}$ in (16) follows from the fact that an interval of length L_k contains exactly (L_k/ϵ) members of the set $\mathcal{X}$, and the multinomial coefficient $n!/(h_1! \cdots h_K!)$ counts the number of arrangements of n objects into K boxes each containing $h_1, \ldots, h_K$ objects, respectively.

Although the final form (17) of the parametric complexity is still an exponential sum, we can compute it efficiently. It turns out that (17) is exactly the same as the parametric complexity of a K-valued multinomial, which we studied in (Kontkanen & Myllymäki, 2005). In this work, we derived the recursion

$$\mathcal{R}^n_{h_K} = \mathcal{R}^n_{h_{K-1}} + \frac{n}{K-2} \mathcal{R}^n_{h_{K-2}}, \quad (18)$$

which holds for $K > 2$. It is now straightforward to write a linear-time algorithm based on (18). The computation starts with the trivial case $\mathcal{R}^n_{h_1} \equiv 1$. The case $K = 2$ is a simple sum

$$\mathcal{R}^n_{h_2} = \sum_{h_1 + h_2 = n} \frac{n!}{h_1! h_2!} \left(\frac{h_1}{n} \right)^{h_1} \left(\frac{h_2}{n} \right)^{h_2}, \quad (19)$$

which clearly can be computed in time $\mathcal{O}(n)$. Finally, recursion (18) is applied $K - 2$ times to end up with $\mathcal{R}^n_{h_K}$. The time complexity of the whole computation is $\mathcal{O}(n + K)$.

Having now derived both the maximum likelihood parameters and the parametric complexity, we are now ready to write down the stochastic complexity (5) for the histogram model. We have

$$\begin{aligned} &SC(\mathbf{x}^n \mid C) \\ &\quad = -\log \frac{\prod_{k=1}^{K} \left(\frac{\epsilon \cdot h_k}{L_k \cdot n} \right)^{h_k}}{\mathcal{R}^n_{h_K}} && (20) \\ &\quad = \sum_{k=1}^{K} -h_k (\log(\epsilon \cdot h_k) - \log(L_k \cdot n)) \\ &\qquad + \log \mathcal{R}^n_{h_K}. && (21) \end{aligned}$$

Equation (21) is the basis for measuring the quality of NML histograms, i.e., comparing different cut point sets. It should be noted that as the term $\sum_{k=1}^{K} -h_k \log \epsilon = -n \log \epsilon$ is a constant with respect to C, the value of ϵ does not affect the comparison. In the next section we will discuss how NML-optimal histograms can be found in practice.

4 LEARNING MDL-OPTIMAL HISTOGRAMS

In this section we will describe a dynamic programming algorithm, which can be used to efficiently find both the optimal bin count and the cut point locations. We start by giving the exact definition of the problem. Let $\tilde{\mathcal{C}} \subseteq \mathcal{C}$ denote the *candidate cut point set*, which is the set of cut points we consider in the optimization process. How $\tilde{\mathcal{C}}$ is chosen in practice, depends on the problem at hand. The simplest choice is naturally $\tilde{\mathcal{C}} = \mathcal{C}$, which means that all the possible cut points are candidates. However, if the value of the accuracy parameter ϵ is small or the data range contains large gaps, this choice might not be practical. Another idea would be to define $\tilde{\mathcal{C}}$ to be the set of midpoints of all the consecutive value pairs in the data $\mathbf{x}^n$. This choice, however, does not allow empty bins, and thus the potential large gaps are still problematic.

A much more sensible choice is to place two candidate cut points between each consecutive values in the data. It is straightforward to prove and also intuitively clear that these two candidate points should be placed as close as possible to the respective data points. In this way, the resulting bin lengths are as small as possible, which will produce the greatest likelihood for the data. These considerations suggest that $\tilde{\mathcal{C}}$ should be chosen as

$$\begin{aligned} \tilde{\mathcal{C}} =& (\{\mathbf{x}_j - \epsilon/2 : \mathbf{x}_j \in \mathbf{x}^n\} \cup \{\mathbf{x}_j + \epsilon/2 : \mathbf{x}_j \in \mathbf{x}^n\}) \\ & \setminus \{\mathbf{x}_{\min} - \epsilon/2, \mathbf{x}_{\max} + \epsilon/2\}. \quad (22) \end{aligned}$$

Note that the end points $\mathbf{x}_{\min} - \epsilon/2$ and $\mathbf{x}_{\max} + \epsilon/2$

are excluded from $\tilde{\mathcal{C}}$, since they are always implicitly included in all the cut point sets.

After choosing the candidate cut point set, the histogram density estimation problem is straightforward to define: find the cut point set $C \subseteq \tilde{\mathcal{C}}$ which optimizes the given goodness criterion. In our case the criterion is based on the stochastic complexity (21), and the cut point sets are considered as models. In practical model selection tasks, however, the stochastic complexity criterion itself may not be sufficient. The reason is that it is also necessary to encode the model index in some way, as argued in (Grünwald, 2006). In some tasks, an encoding based on the uniform distribution is appropriate. Typically, if the set of models is finite and the models are of same complexity, this choice is suitable. In the histogram case, however, the cut point sets of different size produce densities which are dramatically different complexity-wise. Therefore, it is natural to assume that the model index is encoded with a uniform distribution over all the cut point sets of the same size. For a K-bin histogram with the size of the candidate cut point set fixed to E, there are clearly $\binom{E}{K-1}$ ways to choose the cut points. Thus, the codelength for encoding them is $\log \binom{E}{K-1}$.

After these considerations, we define the final criterion (or score) used for comparing different cut point sets as

$$\begin{aligned} &B(\mathbf{x}^n \mid E, K, C) \\ &\quad = SC(\mathbf{x}^n \mid C) + \log \binom{E}{K-1} \qquad (23) \\ &\quad = \sum_{k=1}^{K} -h_k \left(\log(\epsilon \cdot h_k) - \log(L_k \cdot n)\right) \\ &\qquad + \log \mathcal{R}_{h_K}^n + \log \binom{E}{K-1}. \qquad (24) \end{aligned}$$

It is clear that there is an exponential number of possible cut point sets, and thus an exhaustive search to minimize (24) is not feasible. However, the optimal cut point set can be found via dynamic programming, which works by tabulating partial solutions to the problem. The final solution is then found recursively.

Let us first assume that the elements of $\tilde{\mathcal{C}}$ are indexed in such a way that

$$\tilde{\mathcal{C}} = \{\tilde{c}_1, \ldots, \tilde{c}_E\}, \ \tilde{c}_1 < \tilde{c}_2 < \cdots < \tilde{c}_E. \qquad (25)$$

We also define $\tilde{c}_{E+1} = \mathbf{x}_{\max} + \epsilon/2$. Denote

$$\hat{B}_{K,e} = \min_{C \subseteq \tilde{\mathcal{C}}} B(\mathbf{x}^{n_e} \mid E, K, C), \qquad (26)$$

where $\mathbf{x}^{n_e} = (\mathbf{x}_1, \ldots, \mathbf{x}_{n_e})$ is the portion of the data falling into interval $[\mathbf{x}_{\min}, \tilde{c}_e]$ for $e = 1, \ldots, E+1$. This means that $\hat{B}_{K,e}$ is the optimizing value of (24) when the data is restricted to $\mathbf{x}^{n_e}$. For a fixed K, $\hat{B}_{K,E+1}$ is clearly the final solution we are looking for, since the interval $[\mathbf{x}_{\min}, \tilde{c}_{E+1}]$ contains all the data.

Consider now a K-bin histogram with cut points $C = (\tilde{c}_{e_1}, \ldots, \tilde{c}_{e_{K-1}})$. Assuming that the data range is restricted to $[\mathbf{x}_{\min}, \tilde{c}_{e_K}]$ for some $\tilde{c}_{e_K} > \tilde{c}_{e_{K-1}}$, we can straightforwardly write the score function $B(\mathbf{x}^{n_{e_K}} \mid E, K, C)$ by using the score function of a $(K-1)$-bin histogram with cut points $C' = (\tilde{c}_{e_1}, \ldots, \tilde{c}_{e_{K-2}})$ as

$$\begin{aligned} &B(\mathbf{x}^{n_{e_K}} \mid E, K, C) \\ &\quad = B(\mathbf{x}^{n_{e_{K-1}}} \mid E, K-1, C') \\ &\qquad - (n_{e_K} - n_{e_{K-1}})(\log(\epsilon \cdot (n_{e_K} - n_{e_{K-1}})) \\ &\qquad - \log((\tilde{c}_{e_K} - \tilde{c}_{e_{K-1}}) \cdot n)) \\ &\qquad + \log \frac{\mathcal{R}_{h_K}^{n_{e_K}}}{\mathcal{R}_{h_{K-1}}^{n_{e_{K-1}}}} + \log \frac{E-K+2}{K-1}, \qquad (27) \end{aligned}$$

since $(n_{e_K} - n_{e_{K-1}})$ is the number of data points falling into the K^{th} bin, $(\tilde{c}_{e_K} - \tilde{c}_{e_{K-1}})$ is the length of that bin, and

$$\log \frac{\binom{E}{K-1}}{\binom{E}{K-2}} = \log \frac{E-K+2}{K-1}. \qquad (28)$$

We can now write the dynamic programming recursion as

$$\begin{aligned} \hat{B}_{K,e} = \min_{e'} \Big\{ &\hat{B}_{K-1,e'} - (n_e - n_{e'}) \cdot (\log(\epsilon \cdot (n_e - n_{e'})) \\ &- \log((\tilde{c}_e - \tilde{c}_{e'}) \cdot n)) \\ &+ \log \frac{\mathcal{R}_{h_K}^{n_e}}{\mathcal{R}_{h_{K-1}}^{n_{e'}}} + \log \frac{E-K+2}{K-1} \Big\}, \qquad (29) \end{aligned}$$

where $e' = K-1, \ldots, e-1$. The recursion is initialized with

$$\hat{B}_{1,e} = -n_e \cdot (\log(\epsilon \cdot n_e) - \log((\tilde{c}_e - (\mathbf{x}_{\min} - \epsilon/2)) \cdot n)), \qquad (30)$$

for $e = 1, \ldots, E+1$. After that, the bin count is always increased by one, and (29) is applied for $e = K, \ldots, E+1$ until a pre-determined maximum bin count $K_{\max}$ is reached. The minimum $\hat{B}_{K,e}$ is then chosen to be the final solution. By constantly keeping track which e' minimizes (29) during the process, the optimal cut point sequence can also be recovered. The time complexity of the whole algorithm is $\mathcal{O}\left(E^2 \cdot K_{\max}\right)$.

5 EMPIRICAL RESULTS

The quality of a density estimator is usually measured by a suitable distance metric between the data generating density and the estimated one. This is often

problematic, since we typically do not know the data generating density, which means that some heavy assumptions must be made. The MDL principle, however, states that the stochastic complexity (plus the codelength for encoding the model index) itself can be used as a goodness measure. Therefore, it is not necessary to use any additional way of assessing the quality of an MDL density estimator. The optimality properties of the NML criterion and the fact that we are able to find the global optimum in the histogram case will make sure that the final result is theoretically valid.

Nevertheless, to demonstrate the behaviour of the NML histogram method in practice we implemented the dynamic programming algorithm of the previous section and ran some simulation tests. We generated data samples of various size from four densities of different shapes (see below) and then used the dynamic programming method to find the NML-optimal histograms. In all the tests, the accuracy parameter ϵ was fixed to 0.1. We decided to use Gaussian finite mixtures as generating densities, since they are very flexible and easy to sample from. The four generating densities we chose and the corresponding NML-optimal histograms using a sample of 10000 data points are shown in Figures 1 and 2. The densities are labeled gm2, gm5, gm6 and gm8, and they are mixtures of 2, 5, 6 and 8 Gaussian components, respectively, with various amount of overlap between the components. From the plots we can see that the NML histogram method is able to capture properties such as multi-modality (all densities) and long tails (gm6). Another nice feature is that the algorithm automatically places more bins to the areas where more detail is needed like the high, narrow peaks of gm5 and gm6.

To see the behaviour of the NML histogram density algorithm with varying amount of data, we generated data samples of various sizes between $100-10000$ from the four generating densities. For each case, we measured the distance between the generating density and the NML-optimal histogram. As the distance measure we used the (squared) Hellinger distance

$$h^2(f,g) = \int (\sqrt{f(x)} - \sqrt{g(x)})^2 dx, \qquad (31)$$

which has often been used in the histogram context before (see, e.g., (Birge & Rozenholc, 2002; Kanazawa, 1993)). The actual values of the Hellinger distance were calculated via numerical integration. The results can be found in Figure 3. The curves are averaged over 10 different samples of each size. The figure shows that the NML histogram density converges to the generating one quite rapidly when the sample size is increased. The shapes of the convergence curves with the four generating densities are also very similar, which is further evidence of the flexibility of the variable-width

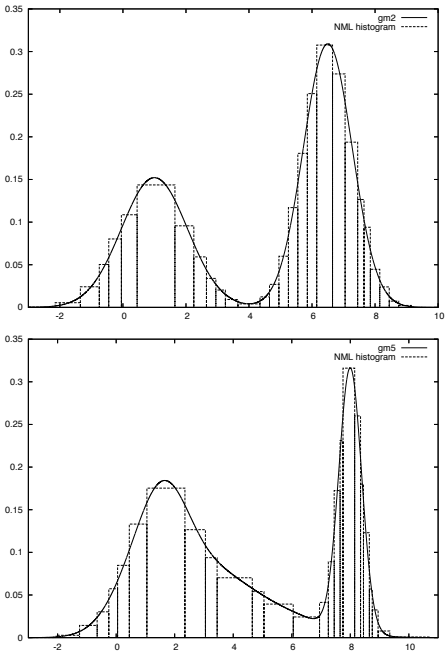

Figure 1: The Gaussian finite mixture densities gm2 and gm5 and the NML-optimal histograms with sample size 10000.

histograms.

To visually see the effect of the sample size, we plotted the NML-optimal histograms against the generating density gm6 with sample sizes 100, 1000 and 10000. These plots can be found in Figure 4. As a reference, we also plotted the empirical distributions of the data samples as a (mirrored) equal-width histograms (the negative y-values). Each bar of the empirical plot has width 0.1 (the value of the accuracy parameter ϵ). When the sample size is 100, the NML histogram algorithm has chosen only 3 bins, and the resulting histogram density is rather crude. However, the small sample size does not justify placing any more bins as can be seen from the empirical distribution. Therefore, we claim that the NML-optimal solution is actually a very sensible one. When the sample size is increased, the bin count is increased and more and more details are captured. Notice that with all the sample sizes, the bin widths of the NML-optimal histograms are strongly variable. It is clear that it would be impossible for any equal-width histogram density estimator to produce such detailed results using the same amount of data.

6 CONCLUSION

In this paper we have presented an information-theoretic framework for histogram density estimation.

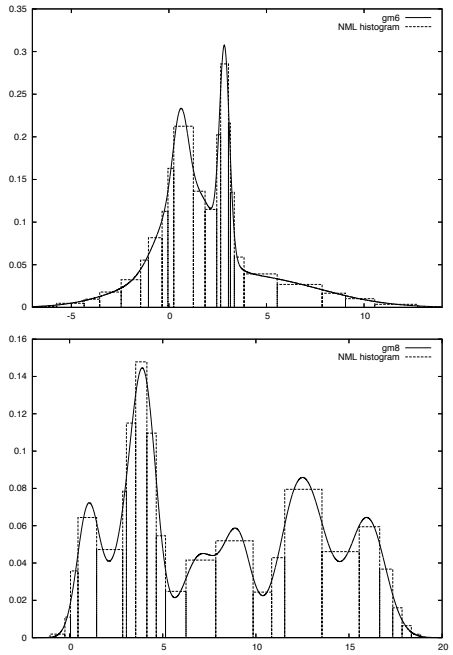

Figure 2: The Gaussian finite mixture densities gm6 and gm8 and the NML-optimal histograms with sample size 10000.

The selected approach based on the MDL principle has several advantages. Firstly, the MDL criterion for model selection (stochastic complexity) has nice theoretical optimality properties. Secondly, by regarding histogram estimation as a model selection problem, it is possible to learn generic, variable-width bin histograms and also estimate the optimal bin count automatically. Furthermore, the MDL criterion itself can be used as a measure of quality of a density estimator, which means that there is no need to assume anything about the underlying generating density. Since the model selection criterion is based on the NML distribution, there is also no need to specify any prior distribution for the parameters.

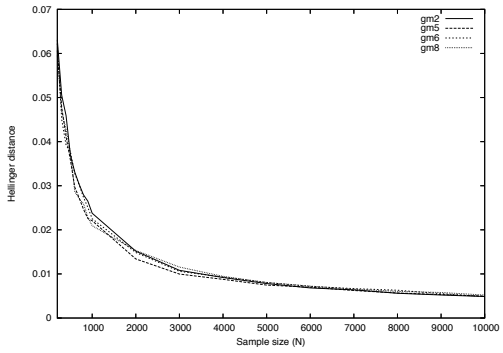

Figure 3: The Hellinger distance between the four generating densities and the corresponding NML-optimal histograms as a function of the sample size.

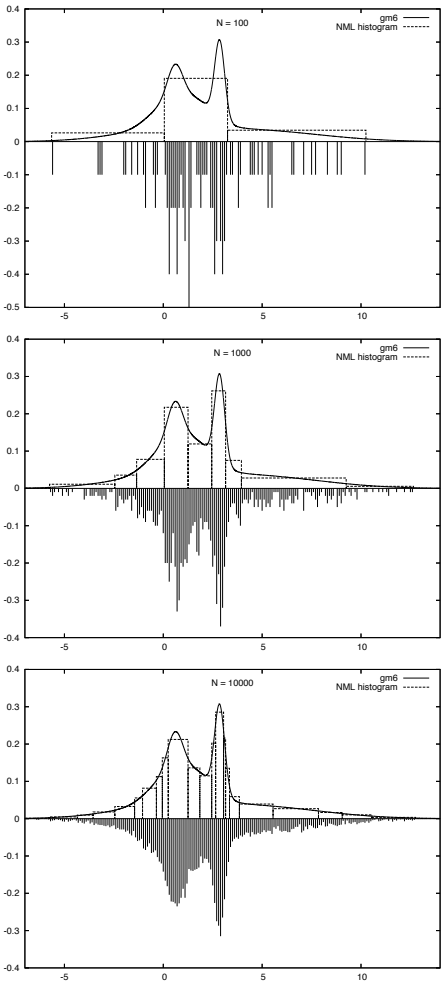

Figure 4: The generating density gm6, the NML-optimal histograms and the empirical distributions with sample sizes 100, 1000 and 10000.

To make our approach practical, we presented an efficient way to compute the value of the stochastic complexity in the histogram case. We also derived a dynamic programming algorithm for efficiently optimizing the NML-based criterion. Consequently, we were able to find the globally optimal bin count and cut point locations in quadratic time with respect to the size of the candidate cut point set.

In addition to the theoretical part, we demonstrated the validity of our approach by simulation tests. In these tests, data samples of various sizes were generated from Gaussian finite mixture densities with highly complex shapes. The results showed that the NML histograms automatically adapt to various kind of densities.

In the future, our plan is to perform an extensive set of empirical tests using both simulated and real data. In these tests, we will compare our approach to other histogram estimators. It is anticipated that the various equal-width estimators will not be performing well in the tests due to the severe limitations of regular histograms. More interesting will be the comparative performance of the density estimator in (Rissanen et al., 1992), which is similar to ours but based on the Bayesian mixture criterion. Theoretically, our version has an advantage at least with small sample sizes.

Another interesting application of NML histograms would be to use them for modeling the class-specific distributions of classifiers such as the Naive Bayes. These distributions are usually modeled with a Gaussian density or a multinomial distribution with equal-width discretization, which typically cannot capture all the relevant properties of the distributions. Although the NML histogram is not specifically tailored for classification tasks, it seems evident that if the class-specific distributions are modeled with high accuracy, the resulting classifier also performs well.

Acknowledgements

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

Balasubramanian, V. (2006). MDL, Bayesian inference, and the geometry of the space of probability distributions. In P. Grünwald, I. Myung, & M. Pitt (Eds.), *Advances in minimum description length: Theory and applications* (pp. 81–98). The MIT Press.

Birge, L., & Rozenholc, Y. (2002, April). *How many bins should be put in a regular histogram.* (Prepublication no 721, Laboratoire de Probabilites et Modeles Aleatoires, CNRS-UMR 7599, Universite Paris VI & VII)

Grünwald, P. (2006). Minimum description length tutorial. In P. Grünwald, I. Myung, & M. Pitt (Eds.), *Advances in minimum description length: Theory and applications* (pp. 23–79). The MIT Press.

Hall, P., & Hannan, E. (1988). On stochastic complexity and nonparametric density estimation. *Biometrika*, *75*(4), 705–714.

Kanazawa, Y. (1993). Hellinger distance and Akaike's information criterion for the histogram. *Statist. Probab. Letters*(17), 293–298.

Kontkanen, P., & Myllymäki, P. (2005). *Analyzing the stochastic complexity via tree polynomials* (Tech. Rep. No. 2005-4). Helsinki Institute for Information Technology (HIIT).

Kontkanen, P., Myllymäki, P., Buntine, W., Rissanen, J., & Tirri, H. (2006). An MDL framework for data clustering. In P. Grünwald, I. Myung, & M. Pitt (Eds.), *Advances in minimum description length: Theory and applications.* The MIT Press.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 445-471.

Rissanen, J. (1987). Stochastic complexity. *Journal of the Royal Statistical Society*, *49*(3), 223–239 and 252–265.

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

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

Rissanen, J. (2005, August). *Lectures on statistical modeling theory.* (Available online at www.mdl-research.org)

Rissanen, J., Speed, T., & Yu, B. (1992). Density estimation by stochastic complexity. *IEEE Transactions on Information Theory*, *38*(2), 315–323.

Roos, T., Myllymäki, P., & Tirri, H. (2005). On the behavior of MDL denoising. In R. G. Cowell & Z. Ghahramani (Eds.), *Proceedings of the 10th international workshop on artificial intelligence and statistics (aistats)* (pp. 309–316). Barbados: Society for Artificial Intelligence and Statistics.

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

Yu, B., & Speed, T. (1992). Data compression and histograms. *Probab. Theory Relat. Fields*, *92*, 195–229.

Paper V

P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, H. Tirri

An MDL Framework for Data Clustering

Advances in Minimum Description Length: Theory and Applications, edited by P. Grünwald, I.J. Myung and M. Pitt. The MIT Press, 2005.

1 An MDL Framework for Data Clustering

Petri Kontkanen
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT) [1]
P. O. Box 9800, FIN-02015 HUT, Finland
petri.kontkanen@hiit.fi, http://www.hiit.fi/petri.kontkanen/

Petri Myllymäki
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
P. O. Box 9800, FIN-02015 HUT, Finland
petri.myllymaki@hiit.fi, http://www.hiit.fi/petri.myllymaki/

Wray Buntine
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
P. O. Box 9800, FIN-02015 HUT, Finland
wray.buntine@hiit.fi, http://www.hiit.fi/wray.buntine/

Jorma Rissanen
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
P. O. Box 9800, FIN-02015 HUT, Finland
jorma.rissanen@hiit.fi, http://www.mdl-research.org/

Henry Tirri
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
P. O. Box 9800, FIN-02015 HUT, Finland
henry.tirri@hiit.fi, http://www.hiit.fi/henry.tirri/

1. HIIT is a joint research institute of University of Helsinki and Helsinki University of Technology.

We regard clustering as a data assignment problem where the goal is to partition the data into several non-hierarchical groups of items. For solving this problem, we suggest an information-theoretic framework based on the minimum description length (MDL) principle. Intuitively, the idea is that we group together those data items that can be compressed well together, so that the total code length over all the data groups is optimized. One can argue that as efficient compression is possible only when one has discovered underlying regularities that are common to all the members of a group, this approach produces an implicitly defined similarity metric between the data items. Formally the global code length criterion to be optimized is defined by using the intuitively appealing universal normalized maximum likelihood code which has been shown to produce optimal compression rate in an explicitly defined manner. The number of groups can be assumed to be unknown, and the problem of deciding the optimal number is formalized as part of the same theoretical framework. In the empirical part of the paper we present results that demonstrate the validity of the suggested clustering framework.

1.1 Introduction

Clustering is one of the central concepts in the field of unsupervised data analysis. Unfortunately it is also a very controversial issue, and the very meaning of the concept "clustering" may vary a great deal between different scientific disciplines (see, e.g., [Jain, Murty, and Flynn 1999] and the references therein). However, a common goal in all cases is that the objective is to find a structural representation of data by grouping (in some sense) similar data items together. In this work we want to distinguish the actual process of grouping the data items from the more fundamental issue of defining a criterion for deciding which data items belong together, and which do not.

In the following we regard clustering as a partitional data assignment or data labeling problem, where the goal is to partition the data into mutually exclusive clusters so that similar (in a sense that needs to be defined) data vectors are grouped together. The number of clusters is unknown, and determining the optimal number is part of the clustering problem. The data are assumed to be in a vector form so that each data item is a vector consisting of a fixed number of attribute values.

Traditionally this problem has been approached by first fixing a distance metric, and then by defining a global goodness measure based on this distance metric — the global measure may for example punish a clustering for pairwise intra-cluster distances between data vectors, and reward it for pairwise inter-cluster distances. However, although this approach is intuitively quite appealing, from the theoretical point of view it introduces many problems.

The main problem concerns the distance metric used: the task of formally describing the desirable properties of a suitable similarity metric for clustering has turned out to be a most difficult task. Commonly used distance metrics include the Euclidean distance and other instances from the Minkowski metric family. However,

although these types of metrics may produce reasonable results in cases where the the underlying clusters are compact and isolated, and the domain attributes are all continuous and have a similar scale, the approach faces problems in more realistic situations [Mao and A.K. 1996].

As discussed in [Kontkanen, Lahtinen, Myllymäki, Silander, and Tirri 2000], non-continuous attributes pose another severe problem. An obvious way to try to overcome this problem is to develop data preprocessing techniques that essentially try to map the problem in the above setting by different normalization and scaling methods. Yet another alternative is to resort to even more exotic distance metrics, like the Mahalanobis distance. However, deciding between alternative distance metrics is extremely difficult, since although the *concept* of a distance metric is intuitively quite understandable, the properties of different distance metrics are far from it [Aggarwal, Hinneburg, and Keim 2001].

A completely different approach to clustering is offered by the *model-based approach*, where for each cluster a data generating function (a probability distribution) is assumed, and the clustering problem is defined as the task to identify these distributions (see, e.g., [Smyth 1999; Fraley and Raftery 1998; Cheeseman, Kelly, Self, Stutz, Taylor, and Freeman 1988]). In other words, the data are assumed to be generated by a finite mixture model [Everitt and Hand 1981; Titterington, Smith, and Makov 1985; McLachlan 1988]. In this framework the optimality of a clustering can be defined as a function of the fit of data with the finite mixture model, not as a function of the distances between the data vectors.

However, the difference between the distance-based and model-based approaches to clustering is not as fundamental as one might think at a first glance. Namely, it is well known that if one, for example, uses the squared Mahalanobis distance in clustering, then this implicitly defines a model-based approach based on Gaussian distributions. A general framework for mapping arbitrary distance functions (or loss functions) to probability distributions is presented in [Grünwald 1998]. The reverse holds of course as well: any explicitly defined probabilistic model can be seen to implicitly generate a distance measure. Consequently, we have two choices: we can either explicitly define a distance metric, which produces an implicitly defined probability distribution, or we can explicitly define a probabilistic model, which implicitly defines a distance metric. We favor the latter alternative for the reasons discussed below.

One of the main advantages of the model-based approach is that the explicit assumptions made correspond to concepts such as independence, linearity, unimodality etc., that are intuitively quite understandable. Consequently, we can argue that constructing a sensible model is easier than constructing a meaningful distance metric. Another important issue is that the modern statistical machine learning community has developed several techniques for automated selection of model complexity. This means that by explicitly defining the model assumptions, one can address the problem of deciding the optimal number of clusters together with the problem of assigning the data vectors to the clusters.

Nevertheless, although the modeling approach has many advantages, it also

introduces some problems. First of all, the finite mixture model implicitly assumes the existence of a hidden clustering variable, the values of which are unknown by definition. Evaluating probabilistic models in this type of an incomplete data case is difficult, and one needs to resort to approximations of theoretically derived model selection criteria. Furthermore, it can also be argued that if the fundamental goal is to find a data partitioning, then it is somewhat counter-intuitive to define the objective of clustering primarily as a model search problem, since clustering is a property of the data, not of the model. Moreover, if one is really interested in the model, and not a partition, then why restrict oneself to a simple finite mixture model? Bayesian or probabilistic networks, for instance, offer a rich family of models that extend the simple mixture model [Lauritzen 1996; Heckerman, Geiger, and Chickering 1995; Cowell, Dawid, Lauritzen, and Spiegelhalter 1999]. A typical survey of users of the Autoclass system [Cheeseman, Kelly, Self, Stutz, Taylor, and Freeman 1988] shows that they start out using clustering, start noticing certain regularities, and then switch over to some custom system. When the actual goal is broader knowledge discovery, model-based clustering is often too simple an approach.

The model-based approach of course implicitly leads to clustering, as the mixture components can be used to compute the probability of any data vector originating from that source. Hence, a mixture model can be used to produce a "soft" clustering where each data vector is assigned to different clusters with some probability. Nevertheless, for our purposes it is more useful to consider "hard" data assignments, where each data vector belongs to exactly one cluster only. In this case we can compute in practice some theoretically interesting model selection criteria, as we shall later see. In addition, it can be argued that this type of hard assignments match more naturally to the human intuition on clustering, where the goodness of a clustering depends on how the data are globally balanced among the different clusterings [Kearns, Mansour, and Ng 1997].

In this paper we propose a model selection criterion for clustering based on the idea that a good clustering is such that one can encode the clustering *together* with the data so that the resulting code length is minimized. In the Bayesian modeling framework this means regarding clustering as a missing data problem, and choosing the clustering (assignment of missing data) maximizing the joint probability. As code lengths and probabilities are inherently linked to each other (see e.g. [Cover and Thomas 1991]), these two perspectives are just two sides of the same coin. But in order to formalize this clustering criterion, we need to explicitly define what we mean by minimal code length / maximal probability. In the Bayesian setting optimality is usually defined with respect to some prior distribution, with the additional assumption that the data actually come from one of the models under consideration.

The main problem with the Bayesian model-based approach for clustering stems from the fact that it implicitly assumes the existence of a latent "clustering variable", the values of which are the missing values that we want to find in clustering. We claim that determining an informative prior for this latent variable

is problematic, as the variable is by definition "hidden"! For example, think of a data set of web log data collected at some WWW site. A priori, we have absolutely no idea of how many underlying clusters of users there exist in the data, or what are the relative sizes of these clusters. What is more, we have also very little prior information about the class-conditional distributions within each cluster: we can of course compute for example the population mean of, say, the age of the users, but does that constitute a good prior for the age within different clusters? We argue that it does not, as what we intuitively are looking for in clustering is discriminative clusters that differ not only from each other, but also from the population as a whole.

The above argument leads to the following conclusion: the Bayesian approach to clustering calls for non-informative (objective) priors that do not introduce any involuntary bias in the process. Formally this can be addressed as a problem for defining so called *reference priors* [Bernardo 1997]. However, current methods for determining this type of priors have technical difficulties at the boundaries of the parameter space of the probabilistic model used [Bernardo 1997]. To overcome this problem, we suggest an information-theoretic framework for clustering, based on the Minimum Description Length (MDL) principle [Rissanen 1978; Rissanen 1987; Rissanen 1996], which leads to an objective criterion in the sense that it is not dependent on any prior distribution, it only uses the data at hand. Moreover, it also has an interpretation as a Bayesian method w.r.t. a worst case prior, and is thus a finite sample variant of the reference prior. It should also be noted that the suggested optimality criterion based on the MDL approach does not assume that the data actually come from the probabilistic model class used for formalizing the MDL principle — this is of course a sensible property in all realistic situations.

In summary, our approach is essentially model-based as it requires an explicit probabilistic model to be defined, no explicit distance metric is assumed. This is in sharp contrast to the information-theoretic approaches suggested in [Gokcay and Principe 2002; Slonim, Friedman, and Tishby 2002], which are essentially distance-based clustering frameworks, where the distance metric is derived from information-theoretic arguments. As discussed above, with respect to the standard model-based Bayesian approach, our approach differs in that the objectivity is approached without having to define an explicit prior for the model parameters.

The clustering criterion suggested here is based on the MDL principle which intuitively speaking aims at finding the shortest possible encoding for the data. For formalizing this intuitive goal, we adopt the modern *normalized maximum likelihood (NML)* coding approach [Shtarkov 1987], which can be shown to lead to a criterion with very desirable theoretical properties (see e.g. [Rissanen 1996; Barron, Rissanen, and Yu 1998; Grünwald 1998; Rissanen 1999; Xie and Barron 2000; Rissanen 2001] and the references therein). It is important to realize that approaches based on either earlier formalizations of MDL, or on the alternative *Minimum Message Length (MML)* encoding framework [Wallace and Boulton 1968; Wallace and Freeman 1987], or on more heuristic encoding schemes (see e.g. [Rissanen and Ristad 1994; Dom 2001; Plumbley 2002; Ludl and Widmer 2002]) do not possess these theoretical properties!

The work reported in [Dom 1995] is closely related to our work as it addresses the problem of segmenting binary strings, which essentially is clustering (albeit in a very restricted domain). The crucial difference is that in [Dom 1995] the NML criterion is used for encoding first the data in each cluster, and the clustering itself (i.e., the cluster labels for each data item) is then encoded *independently*, while in the clustering approach suggested in Section 1.2 all the data (both the data in the clusters plus the cluster indexes) is encoded *together*. Another major difference is that the work in [Dom 1995] concerns binary *strings*, i.e., ordered sequences of data, while we study unordered sets of data. Finally, the computational method used in [Dom 1995] for computing the NML is computationally feasible only in the simple binary case — in Section 1.4 we present a recursive formula that allows us the compute the NML exactly also in more complex, multi-dimensional cases.

This paper is structured as follows. In Section 1.2 we introduce the notation and formalize clustering as a data assignment problem. The general motivation for the suggested information-theoretic clustering criterion is also discussed. In Section 1.3 the theoretical properties of the suggested criterion are discussed in detail. Section 1.4 focuses on computational issues: we show how the suggested MDL clustering criterion can be computed efficiently for a certain interesting probabilistic model class. The clustering criterion has also been validated empirically: illustrative examples of the results are presented and discussed in Section 1.5. Section 1.6 summarizes the main results of our work.

1.2 The clustering problem

1.2.1 Clustering as data partitioning

Let us consider a data set $\mathbf{x}^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ consisting of n outcomes (vectors), where each outcome $\mathbf{x}_j$ is an element of the set $\mathcal{X}$. The set $\mathcal{X}$ consists of all the vectors of the form $(a_1, \ldots, a_m)$, where each variable (or attribute) a_i takes on values on some set that can be either a continuum of real numbers, or a finite set of discrete values. A *clustering* of the data set $\mathbf{x}^n$ is here defined as a partitioning of the data into mutually exclusive subsets, the union of which forms the data set. The number of subsets is a priori unknown. The *clustering problem* is the task to determine the number of subsets, and to decide to which cluster each data vector belongs.

Formally, we can notate a clustering by using a *clustering vector* $y^n = (y_1, \ldots, y_n)$, where y_i denotes the index of the cluster to which the data vector $\mathbf{x}_i$ is assigned to. The number of clusters K is implicitly defined in the clustering vector, as it can be determined by counting the number of different values appearing in y^n. It is reasonable to assume that K is bounded by the size of our data set, so we can define the clustering space Ω as the set containing all the clusterings y^n with the number of clusters being less than n. Hence the clustering problem is now to find from all the $y^n \in \Omega$ the optimal clustering y^n.

For solving the clustering problem we obviously need a global optimization criterion that can be used for comparing clusterings with different number of clusters. On the other hand, as the clustering space Ω is obviously exponential in size, in practice we need to resort to combinatorial search algorithms in our attempt to solve the clustering problem. We return to this issue in Section 1.5. In the following we focus on the more fundamental issue: what constitutes a good optimality criterion for choosing among different clusterings? To formalize this, we first need to explicate the type of probabilistic models we consider.

1.2.2 Model class

Consider a set $\Theta \in \mathbb{R}^d$. A class of parametric distributions indexed by the elements of Θ is called a *model class*. That is, a model class M is defined as the set

$$M = \{P(\cdot|\theta) : \theta \in \Theta\}. \tag{1.1}$$

In the following, we use the simple finite mixture as the model class. In this case, the probability of a single data vector is given by

$$P(\mathbf{x} \mid \theta, M_K) = \sum_{k=1}^{K} P(\mathbf{x} \mid y = k, \theta, M_K) P(y = k \mid \theta, M_K), \tag{1.2}$$

so that a parametric model θ is a weighted mixture of K component models $\theta_1, \ldots, \theta_K$ each determining the local parameters $P(\mathbf{x} \mid y = k, \theta, M_K)$ and $P(y = k \mid \theta, M_K)$. Furthermore, as is usually done in mixture modeling, we assume that the variables $(a_1, \ldots, a_m)$ are locally (conditionally) independent:

$$P(\mathbf{x} \mid y = k, \theta, M_K) = \prod_{i=1}^{m} P(a_i \mid y = k, \theta, M_K). \tag{1.3}$$

The above assumes that the parameter K is fixed. As discussed above, the number of clusters can be assumed to be bounded by the size of the available data set, so in the following we consider the union of model classes $M_1, \ldots, M_n$.

The finite mixture model class is used as an illustrative example in this paper, but it should be noted that the general clustering framework applies of course for other model classes as well. The benefit of the above simple mixture model class is that while it allows arbitrary complex global dependencies with increasing number of components K, from the data mining or data exploration point of view this model class is very appealing as this type of local independence models are very easy to understand and explain.

For the remainder of this paper, we make also the following restricting assumption: we assume that the data are discrete, not continuous, and that the possibly originally continuous variables have been discretized (how the discretization should be done is a difficult problem, and forms a research area that is outside the scope of this paper). One reason for focusing on discrete data is that in this case we can model the domain variables by multinomial distributions without having to make

restricting assumptions about unimodality, normality etc., which is the situation we face in the continuous case. Besides, discrete data are typical to domains such as questionnaire or web log data analysis, and the demand for this type of analysis is increasing rapidly. Moreover, as we shall see in Section 1.4, by using certain computational tricks, in the multinomial case we can compute the theoretically derived objective function presented in the next section exactly, without resorting to approximations. On the other hand, although we restrict ourselves to discrete data in this paper, the information-theoretic framework presented in this paper can be easily extended to cases with continuous variables, or to cases with both continuous and discrete variables, but this is left as a task for future work.

1.2.3 Clustering criterion

Our optimality criterion for clustering is based on information-theoretical arguments, in particular on the Minimum Description Length (MDL) principle [Rissanen 1978; Rissanen 1987; Rissanen 1996]. This also has a perspective from the Bayesian point of view, discussed in more detail in Section 1.3. In the following we try to motivate our approach on a more general level.

Intuitively, the MDL principle aims at finding the shortest possible encoding for the data, in other words the goal is to find the most compressed representation of the data. Compression is possible by exploiting underlying regularities found in the data — the more regularities found, the higher the compression rate. Consequently, the MDL optimal encoding has found all the available regularities in the data; if there would be an "unused" regularity, this could be used for compressing the data even further.

What does this mean in the clustering framework? We suggest the following criterion for clustering: *the data vectors should be partitioned so that the vectors belonging to the same cluster can be compressed well together.* This means that those data vectors that obey the same set of underlying regularities are grouped together. In other words, the MDL clustering approach defines an implicit multilateral distance metric between the data vectors.

How to formalize the above intuitively motivated MDL approach for clustering? Let us start by noting the well-known fact about the fundamental relationship between codes and probability distributions: for every probability distribution P, there exists a code with a code length $-\log P(\mathbf{x})$ for all the data vectors $\mathbf{x}$, and for each code there is probability distribution P such that $-\log P(\mathbf{x})$ yields the code length for data vector $\mathbf{x}$ (see [Cover and Thomas 1991]). This means that we can compress a cluster efficiently, if our model class yields a high probability for that set of data. Globally this means that we can compress the full data set $\mathbf{x}^n$ efficiently, if $P(\mathbf{x}^n \mid M)$ is high. Consequently, in the finite mixture framework discussed in Section 1.2.2, we can define the following optimization problem: Find the model class $M_K \in M$ so that $P(\mathbf{x}^n \mid M_K)$ is maximized.

As discussed in the Introduction, the above model-based approach to clustering poses several problems. One problem is that this type of an incomplete data

probability is in this case difficult to compute in practice as the finite mixture formulation (1.3) implicitly assumes the existence of a latent clustering variable y. What is even more disturbing is the fact that actual clustering y^n has disappeared from the formulation altogether, so the above optimization task does not solve the clustering problem as defined in Section 1.2.1. For these reasons, we suggest the following general optimality criterion for finding the optimal clustering $\hat{y}^n$:

$$\hat{y}^n = \arg\max_{y^n} P(\mathbf{x}^n, y^n \mid M), \tag{1.4}$$

where M is a probabilistic model class.

It is important to notice here is that in this suggested framework, optimality with respect to clustering is defined as a relative measure that depends on the chosen model class M. We see no alternative to this: any formal optimality criterion is necessarily based on some background assumptions. We consider it very sensible that in this framework the assumptions must be made explicit in the definition of the probabilistic model class M. In addition to this, although we in this approach end up with an optimal data partitioning $\hat{y}^n$, which was our goal, we can in this framework also compare different model classes with respect to the question of how well they compress and partition the data.

From the coding point of view, definition (1.4) means the following: If one uses separate codes for encoding the data in different clusters, then in order to be able to decode the data, one needs to send with each vector the index of the corresponding code to be used. This means that we need to encode not only the data $\mathbf{x}^n$, but also the clustering y^n, which is exactly what is done in (1.4).

Definition (1.4) is incomplete in the sense that it does not determine how the joint data probability should be computed with the help of the model class M. In the Bayesian framework this would be done by integrating over some prior distribution over the individual parameter instantiations on M:

$$P(\mathbf{x}^n, y^n \mid M) = \int P(\mathbf{x}^n, y^n \mid \theta, M) P(\theta \mid M) d\theta. \tag{1.5}$$

As discussed in the Introduction, in the clustering framework very little can be known about the model parameters a priori, which calls for objective (non-informative) priors. Typical suggestions are the uniform prior, and the Jeffreys prior. In our discrete data setting, the basic building block of the probability in (1.4) is the Multinomial distribution. As the values of the clustering variable are in our approach based on (1.4) known, not hidden, it follows that instead of a sum as in (1.2), the joint likelihood of a data vector $\mathbf{x}, y$ reduces to a product of Multinomials. This means that the (conjugate) prior $P(\theta)$ is a product of Dirichlet distributions. In the case of the uniform prior, all the individual Dirichlet distributions have all the hyperparameters set to 1. As shown in [Kontkanen, Myllymäki, Silander, Tirri, and Grünwald 2000], the Jeffreys prior is in this case

given by

$$\theta \sim \mathrm{Di}\left(\frac{1}{2}\left(\sum_{i=1}^{m}(n_i-1)+1\right),\ldots,\frac{1}{2}\left(\sum_{i=1}^{m}(n_i-1)+1\right)\right) \times \prod_{i=1}^{m}\prod_{k=1}^{K}\mathrm{Di}\left(\frac{1}{2},\ldots,\frac{1}{2}\right), \quad (1.6)$$

where n_i denotes the number of values of variable a_i, K is the number of clusters, and m is the number of variables (not counting the clustering variable y). Yet another possibility is to use the prior suggested in [Buntine 1991], which is given by

$$\theta \sim \mathrm{Di}\left(\frac{r}{K},\ldots,\frac{r}{K}\right)\prod_{i=1}^{m}\prod_{k=1}^{K}\mathrm{Di}\left(\frac{r}{Kn_i},\ldots,\frac{r}{Kn_i}\right). \quad (1.7)$$

Properties of this prior are discussed in [Heckerman, Geiger, and Chickering 1995]. Parameter r is the so called *equivalent sample size (ESS)* parameter that needs to be determined. Unfortunately, as can be seen in Section 1.5, the value of the equivalent sample size parameter affects the behavior of the resulting clustering criterion a great deal, and we are aware of no disciplined way for automatically determining the optimal value.

In the next section we discuss an information-theoretic framework where the joint probability of the data and the clustering can be determined in an objective manner without an explicit definition of a prior distribution for the model parameters. Section 1.4 (see Equation (1.24)) shows how this framework can be applied for computing the clustering criterion (1.4). In Section 1.5 this information-theoretic approach to clustering is studied empirically and compared to the Bayesian alternatives.

1.3 Stochastic complexity and the minimum description length principle

The information-theoretic *Minimum Description Length (MDL)* principle developed by Rissanen [Rissanen 1978; Rissanen 1987; Rissanen 1989; Rissanen 1996] offers a well-founded theoretical framework for statistical modeling. Intuitively, the main idea of this principle is to represent a set of models (model class) by a single model imitating the behavior of any model in the class. Such representative models are called *universal*. The universal model itself does not have to belong to the model class as often is the case.

The MDL principle is one of the *minimum encoding* approaches to statistical modeling. The fundamental goal of the minimum encoding approaches is *compression of data*. That is, given some sample data, the task is to find a description or *code* of it such that this description uses the least number of symbols, less than other codes and less than it takes to describe the data literally. Intuitively speaking,

in principle this approach can be argued to produce the best possible model of the problem domain, since in order to be able to produce the most efficient coding of data, one must capture all the regularities present in the domain.

The MDL principle has gone through several evolutionary steps during the last two decades. For example, the early realization of the MDL principle (the two-part code MDL [Rissanen 1978]) takes the same form as the *Bayesian information criterion (BIC)* [Schwarz 1978], which has led some people to incorrectly believe that these two approaches are equivalent. The latest instantiation of MDL discussed here is *not* directly related to BIC, but to the formalization described in [Rissanen 1996]. The difference between the results obtained with the "modern" MDL and BIC can be in practice quite dramatic, as demonstrated in [Kontkanen, Buntine, Myllymäki, Rissanen, and Tirri 2003].

Unlike some other approaches, like for example Bayesianism, the MDL principle does not assume that the model class is correct (technically speaking, in the Bayesian framework one needs to define a prior distribution over the model class M, yielding a zero probability to models θ outside this set). It even says that there is no such thing as a true model or model class, as acknowledged by many practitioners. This becomes apparent in Section 1.3.3: the MDL principle can be formalized as a solution to an optimization problem, where the optimization is done over all imaginable distributions, not just over the parametric model class M. Consequently, the model class M is used only as a technical device for constructing an efficient code, and no prior distribution over the set M is assumed.

1.3.1 Stochastic complexity as normalized maximum likelihood

The most important notion of MDL is the *Stochastic Complexity (SC)*. Intuitively, stochastic complexity is defined as the shortest description length of a given data relative to a model class. In the following we give the definition of stochastic complexity, before giving its theoretical justification in the next subsection.

Let $\hat{\theta}(\mathbf{x}^n)$ denote the *maximum likelihood* estimate of data $\mathbf{x}^n$, i.e.,

$$\hat{\theta}(\mathbf{x}^n) = \arg\max_{\theta \in \Theta} \{P(\mathbf{x}^n | \theta, M)\}. \tag{1.8}$$

The stochastic complexity is then defined in terms of the likelihood evaluated at its maximum $P(\mathbf{x}^n \mid \theta, M)|_{\theta=\hat{\theta}(\mathbf{x}^n)}$ as

$$\begin{aligned} SC(\mathbf{x}^n \mid M) &= -\log \frac{P(\mathbf{x}^n \mid \theta, M)|_{\theta=\hat{\theta}(\mathbf{x}^n)}}{R_M^n} \\ &= -\log P(\mathbf{x}^n \mid \theta, M)|_{\theta=\hat{\theta}(\mathbf{x}^n)} + \log R_M^n, \end{aligned} \tag{1.9}$$

where R_M^n is given by

$$R_M^n = \sum_{\mathbf{x}^n} P(\mathbf{x}^n \mid \theta, M)|_{\theta=\hat{\theta}(\mathbf{x}^n)}, \tag{1.10}$$

and the sum goes over all the possible data matrices of length n. The term $\log R_M^n$

is called the *regret* and since it depends on the length of data, not the data itself, it can be considered as a normalization term, and the distribution in (1.9) is called the *normalized maximum likelihood (NML)* distribution proposed for finite alphabets in [Shtarkov 1987]. The definition (1.9) is intuitively very appealing: every data matrix is modeled using its own maximum likelihood (i.e. best fit) model, and then a penalty for the complexity of the model class M is added to normalize the distribution.

1.3.2 Normalized maximum likelihood as a two-part code

A two-part code is such that one first encodes the model to be used for coding, and then the data with the help of the model. Consequently, the total code length consists of a sum of two terms, both of which are lengths of codes produced by proper codes. In its definitional form in (1.9), NML is not a two-part code because the (minus) log regret term is subtracted from the first term.

To make this a two part code, we use the following interpretation: the statistical event $\mathbf{x}^n$ can be broken down into two parts: the first part is the event $\hat{\theta}(\mathbf{x}^n)$ which means we are supplied with the data maximum likelihood but not the data itself; the second part is the event $\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n)$ which then supplies us with the full data. For a simple one dimensional Gaussian model, this means receiving the sample mean first, and then secondly receiving the full set of data points. For distributions with sufficient statistics, the first part $\hat{\theta}(\mathbf{x}^n)$ is generally all that is interesting in the data anyway!

The stochastic complexity (1.9) can now be manipulated as follows:

$$\begin{aligned} SC(\mathbf{x}^n \mid M) &= -\log \frac{P(\mathbf{x}^n, \hat{\theta}(\mathbf{x}^n) \mid \theta, M)\Big|_{\theta=\hat{\theta}(\mathbf{x}^n)}}{R_M^n} \\ &= -\log P(\hat{\theta}(\mathbf{x}^n)|n, M) - \log P(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), \theta, M)\Big|_{\theta=\hat{\theta}(\mathbf{x}^n)} \end{aligned} \tag{1.11}$$

where

$$P(\hat{\theta}(\mathbf{x}^n)|n, M) = \frac{P(\hat{\theta}(\mathbf{x}^n) \mid \theta, M)\Big|_{\theta=\hat{\theta}(\mathbf{x}^n)}}{\sum_{\hat{\theta}} P(\hat{\theta}(\mathbf{x}^n) = \hat{\theta} \mid \theta, M)\Big|_{\theta=\hat{\theta}(\mathbf{x}^n)}}. \tag{1.12}$$

The normalizing term of $P(\hat{\theta}(\mathbf{x}^n)|n, M)$ is just the regret (1.10) with the summation rearranged.

The NML version of stochastic complexity is now a two-part code. The first part encodes the maximum likelihood value $\hat{\theta}(\mathbf{x}^n)$ according to the prior

$$P(\hat{\theta}(\mathbf{x}^n)|n, M) \;\propto\; \max_{\theta} P(\hat{\theta}(\mathbf{x}^n) \mid \theta, M)\ . \tag{1.13}$$

Thus the parameter space Θ has been discretized to values achieving a maximum likelihood for some sample of size n, and the prior distributed so each has its

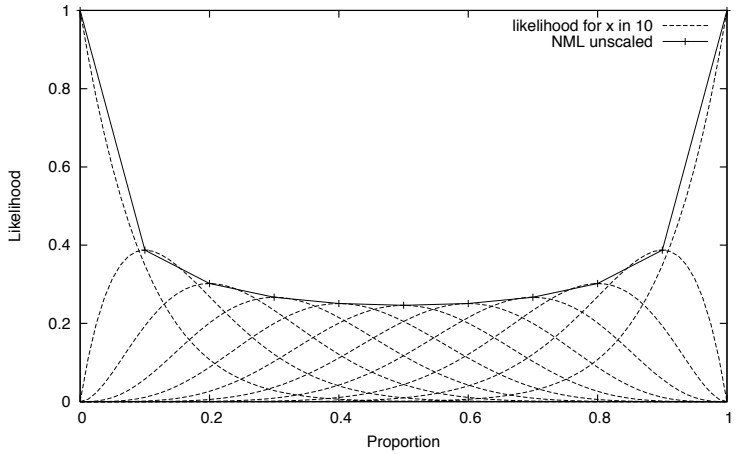

Figure 1.1 Likelihood curves for K=2, n=10.

highest possible likelihood. This construction is given in Figure 1.1 for the binomial model with sample size $n = 10$. Each dashed curve gives a likelihood for a different number of, say 1's, in the data, yielding 11 curves in all. The stochastic complexity is then computed for $\hat{\theta} = 0, 1/10, 2/10, \ldots, 9/10, 1$, which before scaling by regret yields the solid curve. NML at the discretized points $\hat{\theta}$ for different sample sizes $n = 2, 4, \ldots, 128$ is given in Figure 1.2. Notice since this is a discrete distribution, the probability at the points sums to one, and thus the values decrease on average as $1/(n + 1)$.

The second part of the two-part code encodes the *remainder of the data* given the maximum likelihood value $\hat{\theta}(\mathbf{x}^n)$ already encoded. Thus this is no longer a standard sequential code for independent data. In the one dimensional Gaussian case, for instance, it means the sample mean is supplied up front and then the remainder of the data follows with a dependence induced by the known mean.

The ingenious nature of the NML construction now becomes apparent: *One is in effect using a two part code to encode the data, yet no data bits have been wasted in defining the parameters θ since these also form part of the data description itself.* This two part code appears to be a complex codelength to construct in pieces. However, *one computes this two-part codelength without having to explicitly compute the codelengths for the two parts.* Rather, the regret is computed once and for all for the model class and the regular sequential code for data $(-\log P(\mathbf{x}^n \mid \theta, M))$ is the basis for the computation.

One is tempted to continue this construction to interpret $P(\hat{\theta}|n, M)$ based on some reduction to a prior $P(\theta|M)$ over the full parameter space Θ, not just the maximum likelihood values for samples of size n. But this is apparently not possible in the general case. Moreover, in many cases no unique such prior exists. For typical exponential family distributions, for instance, the dimensionality of $P(\hat{\theta}|n, M)$ is

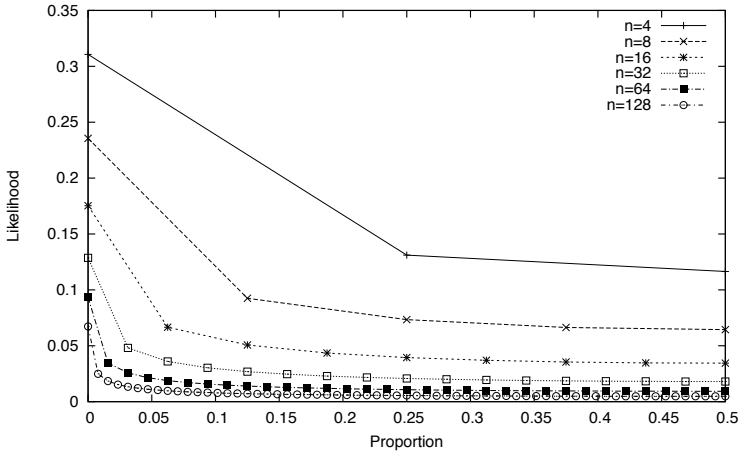

Figure 1.2 NML distribution for K=2, different n.

less than $P(\theta|M)$ and no unique prior will exist except in a limiting sense when $n \to \infty$. We discuss this situation next.

1.3.3 Normalized maximum likelihood as an optimization problem

There have been a number of different alternatives for NML proposed in the literature over the years. We compare some of these here. They provide us with theoretical counterparts to our experimental results.

There are different standards one might use when comparing codelengths on data.

Best case: The optimal possible value for encoding the data $\mathbf{x}^n$ according to model M is $\log 1/P(\mathbf{x}^n|\hat{\theta}(\mathbf{x}^n), M)$, which is unrealizable because $\hat{\theta}$ needs to be known.

Average of best case: Assuming a particular θ for model M holds, the average of the best case is $E_{P(\mathbf{x}^n|\theta,M)} \log 1/P(\mathbf{x}^n|\hat{\theta}(\mathbf{x}^n), M)$.

Barron, Rissanen, and Yu [1998] summarize various optimization problems with respect to these. First, one needs the codelength that will actually be used, $Q(\mathbf{x}^n)$, which is the length we are optimizing.

NML is sometimes derived as the following: find a $Q(\cdot)$ minimizing the worst case (for $\mathbf{x}^n$) increase over the best case codelength for $\mathbf{x}^n$:

$$\min_{Q(\cdot)} \max_{\mathbf{x}^n} \log \frac{P(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), M)}{Q(\mathbf{x}^n)}. \tag{1.14}$$

Stochastic complexity $SC(\mathbf{x}^n)$ is the minimizing distribution here [Shtarkov 1987]. Notice this requires no notion of truth, only a model family used in building a code.

A related definition is based on the average best case codelength for θ: Find a $Q(\cdot)$

minimizing the worst case (for θ) increase over the average best case codelength for θ,

$$\begin{aligned}
&\min_{Q(\cdot)} \max_{\theta} E_{P(\mathbf{x}^n|\theta,M)} \log \frac{P(\mathbf{x}^n|\hat{\theta}(\mathbf{x}^n),M)}{Q(\mathbf{x}^n)} \\
&= \min_{Q(\cdot)} \max_{P(\theta|M)} E_{P(\theta|M)} E_{P(\mathbf{x}^n|\theta,M)} \log \frac{P(\mathbf{x}^n|\hat{\theta}(\mathbf{x}^n),M)}{Q(\mathbf{x}^n)} \\
&= \max_{P(\theta|M)} E_{P(\theta|M)} E_{P(\mathbf{x}^n|\theta,M)} \log \frac{P(\mathbf{x}^n|\hat{\theta}(\mathbf{x}^n),M)}{P(\mathbf{x}^n|M)} \\
&= \log R_M^n - \min_{P(\theta|M)} KL\left(P(\mathbf{x}^n|M) \| SC(\mathbf{x}^n|M)\right). \qquad (1.15)
\end{aligned}$$

The first step is justified changing a maximum $\max_\theta$ into $\max_{P(\theta|M)} E_{P(\theta|M)}$, the second step is justified using minimax and maximin equivalences [Barron, Rissanen, and Yu 1998] since

$$P(\mathbf{x}^n|M) = \arg\min_{Q(\mathbf{x}^n)} E_{P(\mathbf{x}^n,\theta|M)} \log \frac{P(\mathbf{x}^n|\hat{\theta}(\mathbf{x}^n),M)}{Q(\mathbf{x}^n)}, \qquad (1.16)$$

and the third step comes from the definition of $SC(\mathbf{x}^n|M)$.

This optimization then yields the remarkable conclusions for the average best case:

- Finding a $Q(\mathbf{x}^n)$ minimizing the worst case over θ is equivalent to finding a prior $P(\theta|M)$ maximizing the average over θ, although the prior found may not be unique. One could call this a "worst-case Bayesian" analysis that is similar to the so-called *reference prior* analysis of Bernardo [Bernardo 1997]: a $\max_{P(\theta|M)}$ term has been added to a standard formula to minimize a posterior expected cost. However, it applies to the finite sample case, and thus is surely more realistic in practice.
- The minimizing $Q(\mathbf{x}^n)$ must be a valid marginal $P(\mathbf{x}^n|M)$ for some joint $P(\theta|M)P(\mathbf{x}^n|\theta,M)$. Otherwise it is the closest in Kullback-Leibler divergence to the NML distribution. If for some prior $P(\theta|M)$ the induced marginal $P(\mathbf{x}^n|M)$ approaches the NML, then that prior must approach the optimal. Thus NML provides the gold standard for this average case.
- In particular, for exponential family distributions the likelihood for the sufficient statistics of the data and the likelihood for their maximum likelihood value $\hat{\theta}(\mathbf{x}^n)$ are closely related. When the Fisher Information is of full rank, a prior $P(\theta|M)$ with point mass on the set $\{\, \theta : \exists \mathbf{x}^n \text{ such that } \theta = \hat{\theta}(\mathbf{x}^n) \,\}$ can sometimes be found to make the marginal $P(\mathbf{x}^n|M)$ equal to the NML distribution. We claim this holds for the multinomial case. The minimizing $Q(\mathbf{x}^n)$ will thus be the NML in many cases.

Under certain regularity conditions, the optimizing prior approaches Jeffreys prior when $n \to \infty$. Boundaries cause problems here because they mean part of the parameter space is of a lower dimension. For finite n in the case of the multinomial model when the boundaries are included, Xie and Barron [Xie and Barron 2000]

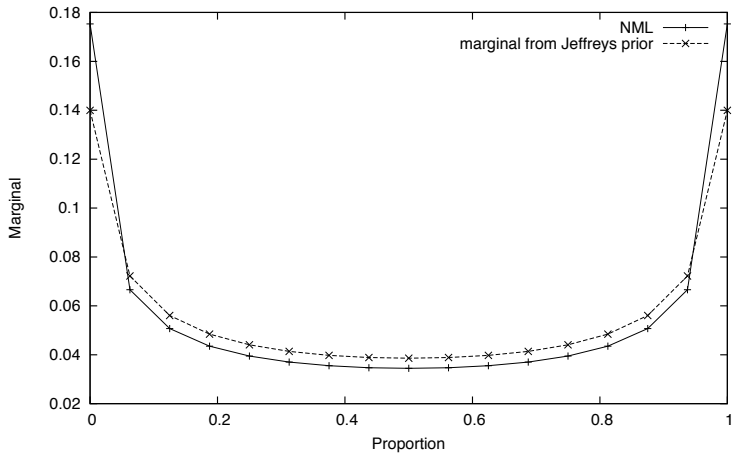

Figure 1.3 Jeffreys prior versus NML as $P(\hat{\theta}|n = 16, M)$ for binomial.

argue for a mixture of Jeffreys priors corresponding to different dimensions being fixed. For the binomial case, this corresponds roughly to mixing a Jeffreys prior with point mass at the two end points ($\theta = 0, 1$). NML versus the Jeffreys prior for the binomial is given in Figure 1.3 for the case when $n = 16$.

For the multinomial for different dimension K and sample size n, NML corresponds closely to Jeffreys prior off the boundaries. The boundaries have significant additional mass. An approximate proportion for Jeffreys prior in the NML distribution is given in Figure 1.4 for the multinomial model with sample sizes $n = 10, \ldots, 1000$ and $K = 2, \ldots, 9$. This records the ratio of NML over the Jeffreys prior at a data point with near equal counts (i.e., off the boundaries). It can be seen that the proportion very slowly rises to 1.0 and for the section here at least is sub-linear in convergence. Xie and Barron use $O(1/n^{1/8})$ for their convergence rate to the Jeffreys prior for the general multinomial. This indicates just how dangerous it is to use the Jeffreys prior as a substitute for the NML distribution in practice.

1.4 Computing the stochastic complexity for multinomial data

1.4.1 One-dimensional case

In the following we instantiate the NML for the one-dimensional multinomial case. Extension to the multi-dimensional model class discussed in Section 1.2.2 is relatively straightforward and is given in Section 1.4.2.

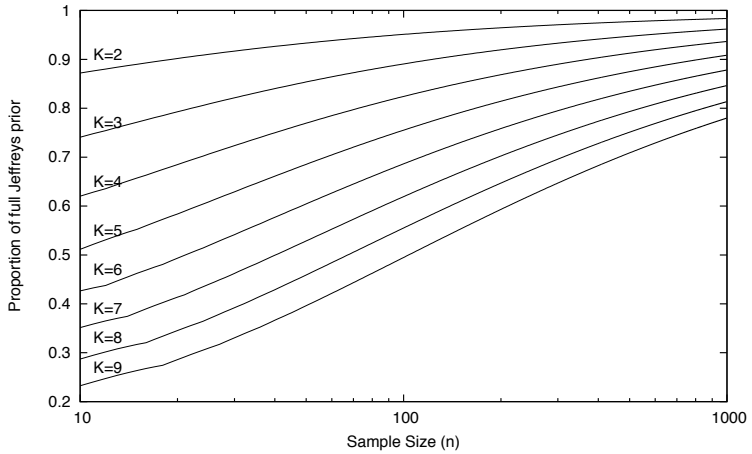

Figure 1.4 Proportion of Jeffreys prior in NML for the multinomial model.

1.4.1.1 Multinomial maximum likelihood

Let us assume that we have a multinomial variable X with K values. The parameter set Θ is then a simplex

$$\Theta = \{(\theta_1, \ldots, \theta_K) : \theta_k \geq 0, \theta_1 + \cdots + \theta_K = 1\}, \tag{1.17}$$

where $\theta_k = P(X = k)$. Under the usual i.i.d. assumption the likelihood of a data set $\mathbf{x}^n$ is given by

$$P(\mathbf{x}^n|\theta) = \prod_{k=1}^{K} \theta_k^{h_k}, \tag{1.18}$$

where h_k is the frequency of value k in $\mathbf{x}^n$. Numbers $(h_1, \ldots, h_K)$ are called the *sufficient statistics* of data $\mathbf{x}^n$. Word "statistics" in this expression means a function of a data and "sufficient" refers to the fact that the likelihood depends on the data only through them.

To instantiate the stochastic complexity (1.9) to the single multinomial case, we need the maximum likelihood estimates of the parameters θ_k, i.e.,

$$\hat{\theta}(\mathbf{x}^n) = (\hat{\theta}_1, \ldots, \hat{\theta}_K) = (\frac{h_1}{n}, \ldots, \frac{h_K}{n}). \tag{1.19}$$

Thus, the likelihood evaluated at the maximum likelihood point is given by

$$P(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n)) = \prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k}. \tag{1.20}$$

1.4.1.2 Multinomial regret

Since the maximum likelihood (1.20) only depends on the sufficient statistics h_k, the regret can be written as

$$R_K^n = \sum_{h_1+\cdots+h_K=n} \frac{n!}{h_1!\cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k}, \tag{1.21}$$

where the summing goes over all the *compositions* of n into K parts, i.e., over all the possible ways to choose non-negative integers $h_1, \ldots, h_K$ so that they sum up to n.

The time complexity of (1.21) is $\mathcal{O}\left(n^{K-1}\right)$, which is easy to see. For example, take case $K = 3$. The regret can be computed in $\mathcal{O}\left(n^2\right)$ time, since we have

$$\begin{aligned} R_K^n &= \sum_{h_1+h_2+h_3=n} \frac{n!}{h_1!h_2!h_3!} \left(\frac{h_1}{n}\right)^{h_1} \left(\frac{h_2}{n}\right)^{h_2} \left(\frac{h_3}{n}\right)^{h_3} \\ &= \sum_{h_1=0}^{n} \sum_{h_2=0}^{n-h_1} \frac{n!}{h_1!h_2!(n-h_1-h_2)!} \cdot \left(\frac{h_1}{n}\right)^{h_1} \left(\frac{h_2}{n}\right)^{h_2} \left(\frac{n-h_1-h_2}{n}\right)^{n-h_1-h_2}. \end{aligned} \tag{1.22}$$

Note that slightly more efficient way for computing the regret would be to sum over *partitions* of n instead of compositions. A (restricted) partition of integer n into K parts is a set of K non-negative integers whose sum is n. For example, compositions $h_1 = 3, h_2 = 2, h_3 = 5$ and $h_1 = 2, h_2 = 5, h_3 = 3$ (with $n = 10$) correspond to the same partition $\{5, 3, 2\}$. Since the maximum likelihood term in (1.21) is clearly different for every partition (but not for every composition), it would be more efficient to sum over the partitions. However, the number of partitions is still $\mathcal{O}\left(n^{K-1}\right)$, so this more complex summing method would not lead to any improvement of the time complexity. Therefore, in order to compute the stochastic complexity in practice, one needs to find better methods. This issue will be addressed below.

1.4.1.3 Recursive formula

A practical method for regret computation is derived via a clever recursion trick. The idea is to find a dependence of R_K^n and regret terms corresponding to a smaller number of values. It turns out that the *double recursive* formula (1.23) derived below offers a solution to this problem. In this formula, R_K^n is represented as a function

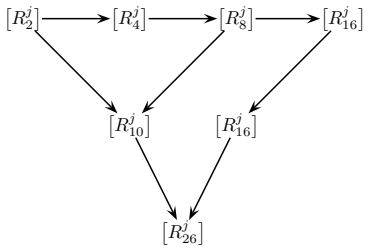

Figure 1.5 Recursive computation of R_{26}^n.

of $R_{K^*}^n$ and $R_{K-K^*}^n$, where K^* can be any integer in $\{1, \ldots, K-1\}$. We have

$$\begin{aligned}
R_K^n &= \sum_{h_1+\cdots+h_K=n} \frac{n!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k} = \sum_{h_1+\cdots+h_K=n} \frac{n!}{n^n} \prod_{k=1}^{K} \frac{h_k^{h_k}}{h_k!} \\
&= \sum_{\substack{h_1+\cdots+h_{K^*}=r_1 \\ h_{K^*+1}+\cdots+h_K=r_2 \\ r_1+r_2=n}} \frac{n!}{n^n} \frac{r_1^{r_1}}{r_1!} \frac{r_2^{r_2}}{r_2!} \left(\frac{r_1!}{r_1^{r_1}} \prod_{k=1}^{K^*} \frac{h_k^{h_k}}{h_k!} \cdot \frac{r_2!}{r_2^{r_2}} \prod_{k=K^*+1}^{K} \frac{h_k^{h_k}}{h_k!} \right) \\
&= \sum_{\substack{h_1+\cdots+h_{K^*}=r_1 \\ h_{K^*+1}+\cdots+h_K=r_2 \\ r_1+r_2=n}} \frac{n!}{n^n} \frac{r_1^{r_1}}{r_1!} \frac{r_2^{r_2}}{r_2!} \left(\frac{r_1!}{h_1! \cdots h_{K^*}!} \prod_{k=1}^{K^*} \left(\frac{h_k}{r_1}\right)^{h_k} \right. \\
&\qquad \left. \cdot \frac{r_2!}{h_{K^*+1}! \cdots h_K!} \prod_{k=K^*+1}^{K} \left(\frac{h_k}{r_2}\right)^{h_k} \right) \\
&= \sum_{r_1+r_2=n} \frac{n!}{r_1! r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \cdot R_{K^*}^{r_1} \cdot R_{K-K^*}^{r_2}. \qquad (1.23)
\end{aligned}$$

This formula can be used in efficient regret computation by applying a combinatoric doubling trick. The procedure goes as follows:

1. Calculate table of R_2^j for $j = 1, \ldots, n$ using the composition summing method (1.21). This can be done in time $\mathcal{O}\left(n^2\right)$.

2. Calculate tables of $R_{2^m}^j$ for $m = 2, \ldots, \lfloor \log_2 K \rfloor$ and $j = 1, \ldots, n$ using the table R_2^j and recursion formula (1.23). This can be done in time $\mathcal{O}\left(n^2 \log K\right)$.

3. Build up R_K^n from the tables. This process also takes time $\mathcal{O}\left(n^2 \log K\right)$.

The time complexity of the whole recursive procedure given above is $\mathcal{O}\left(n^2 \log K\right)$. As an example of this method, say we want to calculate R_{26}^n. The process is illustrated in Figure 1.5. First we form the tables $R_{2^m}^j$ for $m = 1, 2, 3, 4$ and $n = 1, \ldots, N$. Formula (1.23) is then applied to get the tables of R_{10}^j from R_2^j and R_8^j for $j = 1, \ldots, n$. Finally, R_{26}^n can be computed from the tables of R_{16}^j and R_{10}^j.

1.4.2 Multi-dimensional generalization

In this section, we show how to compute NML for the multi-dimensional clustering model class (denoted here by $\mathcal{M}_T$) discussed in Section 1.2.2. Using (1.21), we have

$$SC(\mathbf{x}^n, y^n|\mathcal{M}_T) = -\log\left(\prod_{k=1}^{K}\left(\frac{h_k}{n}\right)^{h_k}\prod_{i=1}^{m}\prod_{k=1}^{K}\prod_{v=1}^{n_i}\left(\frac{f_{ikv}}{h_k}\right)^{f_{ikv}}\right)\cdot\frac{1}{R^n_{\mathcal{M}_T,K}}, \quad (1.24)$$

where h_k is the number of times y has value k in $\mathbf{x}^n$, f_{ikv} is the number of times a_i has value v when $y = k$, and $R^n_{\mathcal{M}_T,K}$ is the regret

$$\begin{aligned} R^n_{\mathcal{M}_T,K} = & \sum_{h_1+\cdots+h_K=n}\ \sum_{f_{111}+\cdots+f_{11n_1}=h_1}\cdots\sum_{f_{1K1}+\cdots+f_{1Kn_1}=h_K}\cdots \\ & \sum_{f_{m11}+\cdots+f_{m1n_m}=h_1}\cdots\sum_{f_{mK1}+\cdots+f_{mKn_m}=h_K}\frac{n!}{h_1!\cdots h_K!}\prod_{k=1}^{K}\left(\frac{h_k}{n}\right)^{h_k} \\ & \cdot\prod_{i=1}^{m}\prod_{k=1}^{K}\frac{h_k!}{f_{ik1}!\cdots f_{ikn_i}!}\prod_{v=1}^{n_i}\left(\frac{f_{ikv}}{h_k}\right)^{f_{ikv}}. \end{aligned} \quad (1.25)$$

Note that we can move all the terms under their respective summation signs, which gives

$$\begin{aligned} R^n_{\mathcal{M}_T,K} = & \sum_{h_1+\cdots+h_K=n}\frac{n!}{h_1!\cdots h_K!}\prod_{k=1}^{K}\left(\frac{h_k}{n}\right)^{h_k} \\ & \cdot\prod_{i=1}^{m}\prod_{k=1}^{K}\sum_{f_{ik1}+\cdots+f_{ikn_i}=h_k}\frac{h_k!}{f_{ik1}!\cdots f_{ikn_i}!}\cdot\prod_{v=1}^{n_i}\left(\frac{f_{ikv}}{h_k}\right)^{f_{ikv}} \\ = & \sum_{h_1+\cdots+h_K=n}\frac{n!}{h_1!\cdots h_K!}\prod_{k=1}^{K}\left(\frac{h_k}{n}\right)^{h_k}\prod_{i=1}^{m}\prod_{k=1}^{K}R^{h_k}_{n_i}, \end{aligned} \quad (1.26)$$

which depends only linearly on the number of variables m, making it possible to compute (1.24) for cases with lots of variables provided that the number of value counts are reasonable small.

Unfortunately, formula (1.26) is still exponential with respect to the number of values $K, n_1, \ldots, n_m$. The situation is especially bad if the number of clusters K is big which often is the case. It turns out, however, that the recursive formula (1.23) can also be generalized to the multi-dimensional case. Proceeding similarly as

in (1.23), we can write

$$
\begin{aligned}
R^n_{\mathcal{M}_T,K} &= \sum_{h_1+\cdots+h_K=n} \left(\frac{n!}{h_1!\cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k} \prod_{i=1}^{m}\prod_{k=1}^{K} R^{h_k}_{n_i} \right) \\
&= \sum_{h_1+\cdots+h_K=n} \left(\frac{n!}{n^n} \prod_{k=1}^{K} \frac{h_k^{h_k}}{h_k!} \prod_{i=1}^{m}\prod_{k=1}^{K} R^{h_k}_{n_i} \right) \\
&= \sum_{\substack{h_1+\cdots+h_{K^*}=r_1 \\ h_{K^*+1}+\cdots+h_K=r_2 \\ r_1+r_2=n}} \left[\frac{n!}{n^n} \frac{r_1^{r_1}}{r_1!} \frac{r_2^{r_2}}{r_2!} \left(\frac{r_1!}{r_1^{r_1}} \prod_{k=1}^{K^*} \frac{h_k^{h_k}}{h_k!} \cdot \frac{r_2!}{r_2^{r_2}} \prod_{k=K^*+1}^{K} \frac{h_k^{h_k}}{h_k!} \right) \right. \\
&\qquad \left. \cdot \prod_{i=1}^{m}\prod_{k=1}^{K^*} R^{h_k}_{n_i} \prod_{k=K^*+1}^{K} R^{h_k}_{n_i} \right], \qquad (1.27)
\end{aligned}
$$

from which we get the result

$$
\begin{aligned}
R^n_{\mathcal{M}_T,K} &= \sum_{\substack{h_1+\cdots+h_{K^*}=r_1 \\ h_{K^*+1}+\cdots+h_K=r_2 \\ r_1+r_2=n}} \left[\frac{n!}{r_1!r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \right. \\
&\qquad \cdot \left(\frac{r_1!}{h_1!\cdots h_{K^*}!} \prod_{k=1}^{K^*} \left(\frac{h_k}{r_1}\right)^{h_k} \prod_{i=1}^{m}\prod_{k=1}^{K^*} R^{h_k}_{n_i} \right) \\
&\qquad \left. \cdot \left(\frac{r_2!}{h_{K^*+1}!\cdots h_K!} \prod_{k=K^*+1}^{K} \left(\frac{h_k}{r_2}\right)^{h_k} \prod_{i=1}^{m}\prod_{k=K^*+1}^{K} R^{h_k}_{n_i} \right) \right] \\
&= \sum_{r_1+r_2=n} \frac{n!}{r_1!r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \cdot R^{r_1}_{\mathcal{M}_T,K^*} \cdot R^{r_2}_{\mathcal{M}_T,K-K^*}. \qquad (1.28)
\end{aligned}
$$

That is, we can calculate multi-dimensional regrets using exactly similar procedures as described in Section 1.4.1.3.

In clustering applications it is typical that the number of clusters K is unknown. Therefore, in order to apply NML for clustering, one needs to evaluate multi-dimensional regrets with varying number of clusters. It follows that the easiest way to use the recursive formula (1.28) is to start with the trivial case $K = 1$, and then always choose $K^* = 1$. The resulting procedure is very simple and as effective as any other, provided that one wants to calculate regrets for the full range $K = 1, \ldots, K_{\max}$. On the other hand, if there is only need to evaluate NML for some fixed K (as is the case if the number of clusters is known), then one should use similar procedures as described in Section 1.4.1.3.

In practice the recursive NML computation for the clustering case goes as follows. The goal is to calculate a $(n \times K_{\max})$ table of multi-dimensional regrets. The procedure starts with the calculation of another array consisting of one-dimensional regrets, since these are needed in (1.28). The size of this array is $(n \times V_{\max})$, where $V_{\max}$ is the maximum of the number of values for the variables $(a_1, \ldots, a_m)$. This array is calculated using (1.23). The time complexity of this step is clearly

$\mathcal{O}\left(V_{\max} \cdot N^2\right)$.

The next step is to determine the starting point for the calculation of the array of multi-dimensional regrets. When $K = 1$, formula (1.26) clearly reduces to

$$R^n_{\mathcal{M}_T,1} = \prod_{i=1}^{m} R^n_{n_i}. \tag{1.29}$$

Another trivial case is $n = 0$, which gives

$$R^0_{\mathcal{M}_T,K} = 1, \tag{1.30}$$

for all K. After that, the calculation proceeds by always increasing n by one, and for each fixed n, increasing K by one up to the maximum number of clusters wanted.

The interesting thing is that although the multi-dimensional regret formula (1.26) is rather complicated, the described procedure never uses it directly. The only things needed are the trivial starting cases $K = 1$ and $n = 0$, and the recursive formula (1.28). It follows that the calculation of multi-dimensional regrets is computationally as effective as in the single-dimensional case, which is a rather surprising but important fact.

1.5 Empirical results

1.5.1 Clustering scoring methods

We have presented a framework for data clustering where the validity of a clustering y^n is determined according to the complete data joint probability in Equation (1.4). Consequently, we obtain different clustering criteria or scoring methods by using different ways for computing this probability. In the following, the following clustering methods were empirically validated:

NML The NML criterion given by Equation (1.9).

UNI The Bayesian criterion given by the marginal likelihood (1.5) over the uniform prior distribution.

JEF The Bayesian criterion given by the marginal likelihood (1.5) over the Jeffreys prior distribution (1.6).

ESS(r) The Bayesian criterion given by the marginal likelihood (1.5) over the prior distribution (1.7). The parameter r is the equivalent sample size required for determining this prior.

The above means that ESS(r) is actually a continuum of methods, as the equivalent sample size can be any positive real number. In the following the following alternatives were tested: ESS(0.01), ESS(0.1), ESS(1.0), ESS(10.0) and ESS(100.0).

1.5.2 Empirical setup

In the following we wish to study empirically how the NML clustering criterion compares with respect to the Bayesian scores UNI, JEF and ESS(r). The problem is now to find an empirical setup where these different criteria can be compared objectively. However, this turns out to be a most difficult task. Namely, at first sight it seems that an objective empirical scenario can be obtained by the following setup:

1. Choose randomly K probability distributions $P(\mathbf{x} \mid \Theta_1), \ldots, P(\mathbf{x} \mid \Theta_K)$.
2. i:=1.
3. Generate data $\mathbf{x}^n$ by repeating the following procedure n times:
   (a) Choose a random number z_i between 1 and K.
   (b) Draw randomly a data vector $\mathbf{x}_i$ from distribution $P(\mathbf{x} \mid \Theta_{z_i})$.
   (c) i:=i+1.
4. Cluster the generated data $\mathbf{x}^n$ in order to get a clustering y^n.
5. Validate the clustering by comparing y^n and the "ground truth" z^n.

We claim that the above procedure has several major weaknesses. One issue is that the setup obviously requires a search procedure in step 4, as the clustering space is obviously exponential in size. However, any heuristic search algorithm chosen for this purpose may introduce a bias favoring some of the criteria.

More importantly, one can argue that the "original" clustering z^n is not necessarily the goal one should aim at: Consider a case where the data was generated by a 10-component mixture model, where two of the components are highly overlapping, representing almost the same probability distribution. We claim that in this case a sensible clustering method should produce a clustering with 9 clusters, not 10! On the other hand, consider a case where all the 10 component distributions are not overlapping, but only one sample has been drawn from each of the 10 components. We argue that in this case a sensible clustering criterion should suggest a relatively small number of clusters, say 1 or 2, instead of the "correct" number 10, since with small sample sizes the variation in the data could not possibly justify the use of so many clusters (meaning a high number of parameters).

This means that the above scenario with artificial data makes only sense if the mixture components are non-overlapping, and the amount of data is substantial. Obviously it can now be argued that this unrealistic situation hardly resembles real-world clustering problems, so that the results obtained in this way would not be very relevant. What is more, if the data are generated by a finite mixture of distributions, which means that the local independence assumptions we made in Section 1.2.2 do indeed hold, then this setup favors the Bayesian approach as in this unrealistic case the marginal likelihood criterion is also minimax optimal. A more realistic setup would of course be such that the assumptions made would not hold, and the data would *not* come from any of the models in our model class.

The above scenario can be modified to a more realistic setting by changing the data generating mechanism so that the assumptions made do not hold any more. One way to achieve this goal in our local independence model case would be to add dependencies between the variables. However, this should be done in such a manner that the dependencies introduced are sensible in the sense that such dependencies exist in realistic domains. This is of course a most difficult task. For this reason, in the set of experiments reported here we used real-world data that were gathered in a controlled manner so that the above testing procedure could be used although reality was used as a data generating mechanism instead of a manually constructed mixture model. Before describing the data, let us have a look at the actual clustering procedure used in the experiments.

1.5.3 The search algorithm

For the actual clustering algorithm, we studied several alternatives. The best results were obtained with a simple stochastic greedy algorithm, where the number of clusters K was first fixed, and then the following procedure repeated several times:

1. Choose a random initial data assignment.
2. Choose a random data vector.
3. Move the chosen data vector to the cluster optimizing locally the clustering score.
4. If converged, stop. Otherwise, go to step 2.

This procedure was repeated with all the possible values for K, and with all the clustering scoring methods listed in Section 1.5.1. At the end, all the clusterings of different size, produced by all the runs with all the clustering methods, were put together into a large pool of candidate clusterings. Finally, all the candidate clusterings were evaluated by using all the clustering criteria. The purpose of this procedure was to prevent the effect of chance between individual runs of the stochastic search algorithm with different criteria. It should be noted, however, that in our experiments almost all the best clusterings were found using NML as the clustering score. We believe that this tells something important about the shape of the search space with different clustering criteria, and this interesting issue will be studied in our future research.

1.5.4 The data

In this set of experiments, the data consisted of measured signal strength values of radio signals originating from eight WLAN access points (transmitters) located in different parts of our laboratory. As the measured signal strength depends strongly on the distance to the transmitting access point, the distribution of the data collected at some fixed point depends on the relative distances of this point and the locations of the eight access points. This means that the measurement distributions at two locations far from each other are very likely to be very different. Furthermore,

as the access points are not affecting each other, the eight measured signals are at any fixed point more or less independent of each other.

Consequently, the data collected in the above manner are in principle similar to artificial data generated by a finite mixture model. Nevertheless, in real-world environments there is always some inherent noise caused by factors such as measurement errors, position and angle of reflecting or damping surfaces, air humidity, presence or absence of people and so on. This means that this type of data resemble artificial data in the sense that the overlap between the component distributions can be controlled by choosing the locations where the measurements are made, but at the same time the data contain realistic type of noise that was not artificially generated.

1.5.5 The results

For this set of experiments, data were gathered at different locations situated as far from each other as possible. This means that the data generating mechanisms were rather different, and partitioning the unlabeled data into clusters corresponding to the measurement locations was relatively easy with all the clustering methods used, if a sufficient number of data was available. However, as we in this setup were able to control the amount of data available, we could study the small sample size behavior of the different clustering scores. A typical example of the behavior of different clustering criteria can be seen in Figures 1.6 and 1.7.

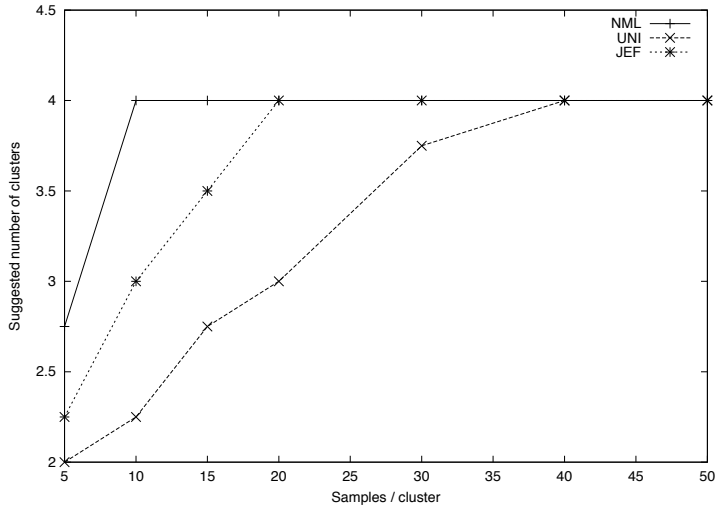

Figure 1.6 An example of the behavior of different clustering scores in the task of finding a four cluster data partitioning, as a function of sample size per cluster.

In Figure 1.6 we see a typical example of how the NML, UNI and JEF clustering

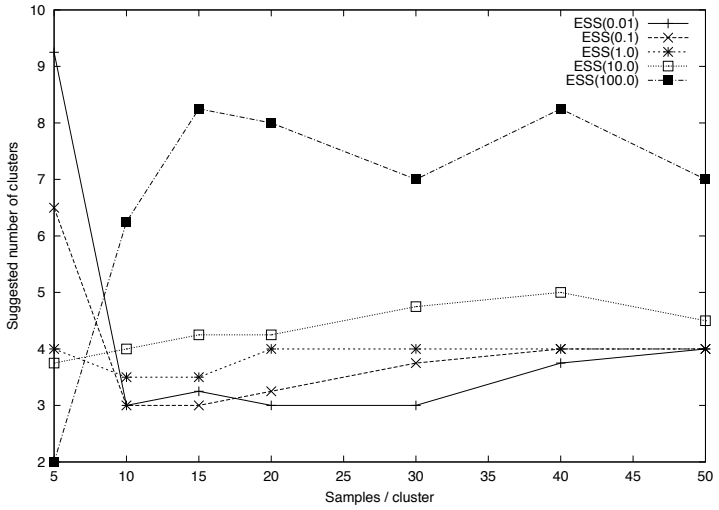

Figure 1.7 An example of the behavior of different ESS clustering scores in the task of finding a four cluster data partitioning, as a function of sample size per cluster.

criteria behave as a function of the sample size. In this case, the correct number of clusters was four (data were gathered at four different positions), and the X-axis gives the number of data vectors collected at each of the 4 locations. The Y-axis gives the number of clusters in the best clustering found with each of the three clustering criteria, where the pool of candidate clusterings were generated as described in Section 1.5.3. In this simple case, whenever the best clustering contained 4 clusters, the actual clustering y^n was perfectly consistent with the way the data were collected, i.e., the clustering suggested was "correct". Obviously, whenever the suggested number of clusters was other than 4, the correct clustering was not found. The values on the Y-axis are averages over several repeats of the sequential procedure consisting of data gathering, construction of the clustering candidate pool and validation of the clustering candidates with different clustering criteria.

From Figure 1.6 we can see that with very small sample sizes (with fewer than 10 samples from each cluster), NML tends to suggest less clusters than there actually is. However, as discussed above, this is a sensible behavior as very little data sets do not justify very complex models. After sample size of 10, the NML always finds the correct number of clusters (and as explained above, also the correct clustering). The behavior of the UNI and JEF scores is very similar, but they need more data in order to find the correct clustering.

The behavior of the ESS scores is rather interesting, as we can see in Figure 1.7. In this particular case, a relatively small equivalent sample size seems to work well: ESS(1) converges rather quickly (after seeing 20 samples per cluster) to the right level. However, the behavior is somewhat counter-intuitive with very small sample

sizes as the suggested number of clusters is first close to 4, then goes down as the sample size increases to 15, after which it goes up again. A similar, but even more disturbing pattern is produced by the ESS scores with small equivalent sample size: with very small samples (under 10 samples per cluster), they tend to suggest clusterings with much too high number of clusters. This of course would lead to poor results in practice.

The ESS scores with a high equivalent sample size increase the suggested number of clusters with increasing data size up to a point, after which they start to converge to the right level. As a matter of fact, after a sufficient number of samples from each cluster, all the clustering criteria typically suggest a clustering identical or very close to the correct clustering. Consequently, this example shows that the interesting differences between the different clustering methods cannot be seen in low-dimensional cases if a large number of data is available. Real world problems are typically very high-dimensional, which means that the amount of data available is always relatively low, which suggests that the small sample size behavior of the clustering criteria observed here is of practical importance.

1.6 Conclusion

We suggested a framework for data clustering based on the idea that a good clustering is such that it allows efficient compression when the data are encoded together with the cluster labels. This intuitive principle was formalized as a search problem, where the goal is to find the clustering leading to maximal joint probability of the observed data plus the chosen cluster labels, given a parametric probabilistic model class.

The nature of the clustering problem calls for objective approaches for computing the required probabilities, as the presence of the latent clustering variable prevents the use of subjective prior information. In the theoretical part of the paper, we compared objective Bayesian approaches to the solution offered by the information-theoretic Minimum Description Length principle, and observed some interesting connections between the Normalized Maximum Likelihood approach and the Bayesian reference prior approach.

To make things more concrete, we instantiated the general data clustering approach for the case with discrete variables and a local independence assumption between the variables, and presented a recursive formula for efficient computation of the NML code length in this case. The result is of practical importance as the amount of discrete data is increasing rapidly (in the form of WWW pages, WWW log data, questionnaires, and so on). Although the approach can be easily extended to more complex cases than the one studied in this paper, we argue that the local independence model is important as the resulting clusters are in this case easy to analyze. It can also be said that the local independence model assumed here is complex enough, as one can obviously model arbitrarily complex distributions by adding more and more clusters.

In the empirical part of the paper we studied the behavior of the NML clustering criterion with respect to the Bayesian alternatives. Although all the methods produced reasonable results in simple low-dimensional cases if sufficient amount of data was available, the NML approach was clearly superior in more difficult cases with insufficient number of data. We believe that this means that NML works better in practical situations where the amount of data available is always vanishingly small with respect to the multi-dimensional space determined by the domain variables.

The difference between NML and the Bayesian approaches was especially clear when compared to the "parameter-free" approaches with either the uniform or the Jeffreys prior. The equivalent sample size prior produced good results if one was allowed to manually choose the ESS parameter, but this of course does not constitute a proper model selection procedure, as no general guidelines for automatically selecting this parameter can be found.

In this paper the clustering framework was restricted to flat, non-overlapping and non-hierarchical clusterings. The approach could be obviously extended to more complex clustering problems by introducing several clustering variables, and by assuming a hierarchical structure between them, but this path was left to be explored in our future research.

Acknowledgements

This work has been supported in part by the Academy of Finland under the projects Cepler and Minos, and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views. The authors wish to thank Michael Lee and Dan Navarro for their encouraging and valuable comments.

Bibliography

Aggarwal, C., A. Hinneburg, and D. Keim (2001). On the surprising behavior of distance metrics in high dimensional space. In J. V. den Bussche and V. Vianu (Eds.), *Proceedings of the Eighth International Conference on Database Theory*, Volume 1973 of *Lecture Notes in Computer Science*, pp. 420–434. Springer-Verlag.

Barron, A., J. Rissanen, and B. Yu (1998). The minimum description principle in coding and modeling. *IEEE Transactions on Information Theory 44*(6), 2743–2760.

Bernardo, J. (1997). Noninformative priors do not exist. *J. Statist. Planning and Inference 65*, 159–189.

Buntine, W. (1991). Theory refinement on Bayesian networks. In B. D'Ambrosio, P. Smets, and P. Bonissone (Eds.), *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 52–60. Morgan Kaufmann Publishers.

Cheeseman, P., J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman (1988). Autoclass: A Bayesian classification system. In *Proceedings of the Fifth International Conference on Machine Learning*, Ann Arbor, pp. 54–64.

Cover, T. and J. Thomas (1991). *Elements of Information Theory*. New York, NY: John Wiley & Sons.

Cowell, R., P. Dawid, S. Lauritzen, and D. Spiegelhalter (1999). *Probabilistic Networks and Expert Systems*. New York, NY: Springer.

Dom, B. (1995). MDL estimation with small sample sizes including an application to the problem of segmenting binary strings using Bernoulli models. Technical Report RJ 9997 (89085), IBM Research Division, Almaden Research Center.

Dom, B. (2001). An information-theoretic external cluster-validity measure. Technical Report RJ 10219, IBM Research.

Everitt, B. and D. Hand (1981). *Finite Mixture Distributions*. London: Chapman and Hall.

Fraley, C. and A. E. Raftery (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal 41*(8), 578–588.

Gokcay, E. and J. Principe (2002). Information theoretic clustering. *IEEE Trans-*

actions on Pattern Analysis and Machine Intelligence 24(2), 158–170.

Grünwald, P. (1998). *The Minimum Description Length Principle and Reasoning under Uncertainty*. Ph. D. thesis, CWI, ILLC Dissertation Series 1998-03.

Heckerman, D., D. Geiger, and D. Chickering (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning 20*(3), 197–243.

Jain, A., M. Murty, and P. Flynn (1999). Data clustering: A review. *ACM Computing Surveys 31*(3), 264–323.

Kearns, M., Y. Mansour, and A. Y. Ng (1997). An information-theoretic analysis of hard and soft assignment methods for clustering. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–97)*, San Francisco, CA, pp. 282–293. Morgan Kaufmann Publishers.

Kontkanen, P., W. Buntine, P. Myllymäki, J. Rissanen, and H. Tirri (2003). Efficient computation of stochastic complexity. In C. Bishop and B. Frey (Eds.), *Proceedings of the Ninth International Conference on Artificial Intelligence and Statistics*, pp. 233–238. Society for Artificial Intelligence and Statistics.

Kontkanen, P., J. Lahtinen, P. Myllymäki, T. Silander, and H. Tirri (2000). Supervised model-based visualization of high-dimensional data. *Intelligent Data Analysis 4*, 213–227.

Kontkanen, P., P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald (2000). On predictive distributions and Bayesian networks. *Statistics and Computing 10*, 39–54.

Lauritzen, S. (1996). *Graphical Models*. Oxford University Press.

Ludl, M.-C. and G. Widmer (2002). Clustering criterion based on minimum length encoding. In T. Elomaa, H. Mannila, and H. Toivonen (Eds.), *Proceedings of the 13th European Conference on Machine Learning*, Volume 2430 of *Lecture Notes in Computer Science*, pp. 258–269. Springer.

Mao, J. and J. A.K. (1996). A self-organizing network for hyperellipsoidal clustering (HEC). *IEEE Transactions on Neural Networks 7*, 16–29.

McLachlan, G. (1988). *Mixture Models: Inference and Applications to Clustering*. New York: Marcel Dekker.

Plumbley, M. (2002). Clustering of sparse binary data using a minimum description length approach. Technical report, Department of Electrical Engineering, Queen Mary, University of London. Unpublished manuscript.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica 14*, 445–471.

Rissanen, J. (1987). Stochastic complexity. *Journal of the Royal Statistical Society 49*(3), 223–239 and 252–265.

Rissanen, J. (1989). *Stochastic Complexity in Statistical Inquiry*. New Jersey: World Scientific Publishing Company.

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

Rissanen, J. (1999). Hypothesis selection and testing by the MDL principle. *Computer Journal 42*(4), 260–269.

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

Rissanen, J. and E. S. Ristad (1994). Unsupervised Classification with Stochastic Complexity. In H. B. et al. (Ed.), *Proceedings of the First US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach*, pp. 171–182. Kluwer Academic Publishers.

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

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

Slonim, N., N. Friedman, and N. Tishby (2002). Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 129–136. ACM Press.

Smyth, P. (1999). Probabilistic model-based clustering of multivariate and sequential data. In D. Heckerman and J. Whittaker (Eds.), *Proceedings of the Seventh International Conference on Artificial Intelligence and Statistics*, pp. 299–304. Morgan Kaufmann Publishers.

Titterington, D., A. Smith, and U. Makov (1985). *Statistical Analysis of Finite Mixture Distributions*. New York: John Wiley & Sons.

Wallace, C. and D. Boulton (1968). An information measure for classification. *Computer Journal 11*, 185–194.

Wallace, C. and P. Freeman (1987). Estimation and inference by compact coding. *Journal of the Royal Statistical Society 49*(3), 240–265.

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

Paper VI

P. Kontkanen and P. Myllymäki

An Empirical Comparison of NML Clustering Algorithms

In Proceedings of the 2008 International Conference on Information Theory and Statistical Learning (ITSL-08).

An Empirical Comparison of NML Clustering Algorithms

Petri Kontkanen and Petri Myllymäki
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
University of Helsinki and Helsinki University of Technology
Helsinki, Finland

Abstract - *Clustering can be defined as a problem of partitioning a given data into non-hierarchical groups of items. In our previous work, we suggested an information-theoretic criterion for defining the goodness of a clustering of data. The basic idea behind this framework is to optimize the total code length over the data by encoding together data items belonging to the same cluster. Formally the global code length criterion to be optimized is defined by using the theoretically and intuitively appealing universal normalized maximum likelihood (NML) code. In this paper, we focus on the optimization aspect of the clustering problem, and study five algorithms that can be used for efficiently searching the exponentially-sized clustering space. The number of clusters is not known beforehand and determining it is part of the optimization process. In the empirical part of the paper we compare the performance of the suggested algorithms using several real-world datasets.*

Keywords: minimum description length, normalized maximum likelihood, clustering, EM algorithm, K-means algorithm

1 Introduction

Although clustering is one of the central concepts in the field of unsupervised data analysis, it is also a very controversial issue, and the very meaning of the concept "clustering" may vary a great deal between different scientific disciplines (see, e.g., [1] and the references therein). However, a common goal in all cases is that the objective is to find a structural representation of data by grouping (in some sense) similar data items together. In the following we regard clustering as a partitional data assignment or data labeling problem, where the goal is to partition the data into mutually exclusive clusters so that similar data vectors are grouped together. The number of clusters is unknown, and determining the optimal number is part of the clustering problem. The data are assumed to be in a vector form so that each data item is a vector consisting of a fixed number of attribute values.

We can now identify two fundamental problems within this framework: how to define the goodness of a clustering (data partitioning) and how to find good clusterings with respect to the chosen scoring criterion. The focus in this paper is on the latter problem.

Traditionally, the scoring problem has been approached by first fixing a distance metric, and then by defining a global goodness measure based on this distance metric, However, although this approach is intuitively quite appealing, from the theoretical point of view it introduces many problems, such as choosing a suitable distance metric and the handling of non-continuous attributes. A completely different approach to clustering is offered by the *model-based approach*, where for each cluster a data generating function (a probability distribution) is assumed, and the clustering problem is defined as the task to identify these distributions (see, e.g., [2, 3, 4]). In other words, the data are assumed to be generated by a finite mixture model [5, 6, 7]. In this framework the optimality of a clustering can be defined as a function of the fit of data with the finite mixture model, not as a function of the distances between the data vectors. See [8] for more discussion on the differences between the traditional and the model-based approaches.

In [8] we proposed a scoring criterion for clusterings, based on the idea that a good clustering is such that one can encode the cluster labels *together* with the data so that the resulting code length is minimized. The clustering criterion suggested was based on the MDL principle [9, 10, 11] which intuitively speaking aims at finding the shortest possible encoding for the data. For formalizing this intuitive goal, we adopt the modern *normalized maximum likelihood (NML)* coding approach [12], which can be shown to lead to a criterion with very desirable theoretical properties (see Section 2 and e.g. [11, 13, 14, 15, 16, 17]). It is important to realize that approaches based on either earlier formalizations of MDL or on more heuristic encoding schemes

(see e.g. [18, 19, 20]) do not possess these theoretical properties.

This paper is a direct continuation of [8], where we introduced the NML clustering approach and derived an efficient algorithm for computing the NML criterion. In the empirical tests of [8] we concentrated on a special data consisting of measured signal strength values of radio signals originating from WLAN access points. In this paper, we will extend this work by using several real-world datasets from the UCI repository [21]. Moreover, we will present and empirically compare five different optimization algorithms that can be used for finding good clusterings with respect to the NML scoring criterion.

This paper is structured as follows. In Section 2 we discuss the basic properties of the MDL framework in general and also shortly review the optimality properties of the NML distribution. In Section 3 we introduce the notation and formalize clustering as a data assignment problem. We also show how the NML criterion can be computed efficiently for the clustering model class. In Section 4, we empirically compare several algorithms for finding good clusterings. Section 5 summarizes the main results of our work.

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 the 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., [11, 13, 16, 17, 22, 23].

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 $\mathbf{x}^n = (x_1, \ldots, x_n)$ be a data sample of n outcomes, where each outcome 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

$$\mathcal{M} = \{P(\cdot \mid \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}, \tag{1}$$

and the set Θ is called a *parameter space*.

One of the most theoretically and intuitively appealing model class selection criteria is the *stochastic complexity*. Denote first 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 [12] is now defined as

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

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

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

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

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

and the term $\log \mathcal{C}(\mathcal{M}, n)$ is called the *(minimax) regret* or *parametric complexity*. The 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 [24] for more discussion on this topic.

The NML distribution (2) has several important theoretical optimality properties. The first one is that NML provides a unique solution to the minimax problem

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

as posed in [12]. The minimizing $\hat{P}$ is the NML distribution, and the minimax regret

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

is given by the parametric complexity $\log \mathcal{C}(\mathcal{M}, n)$. This means that the NML distribution is the *minimax optimal universal model*. The term universal model in this context means that the NML distribution represents (or mimics) the behaviour of all the distributions in the model

class $\mathcal{M}$. 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 [17]. This property states that NML also minimizes

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

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 clustering

Let us assume that our problem domain consists of m discrete variables $X_1, \ldots, X_m$ and that the variable X_i has K_i values. The data $\mathbf{x}^n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ consists of observations $\mathbf{x}_j = (x_{j0}, x_{j1}, \ldots, x_{jm}) \in \mathcal{X}$, where

$$\mathcal{X} = \{1, 2, \ldots, K_1\} \times \cdots \times \{1, 2, \ldots, K_m\}. \tag{8}$$

We assume that the possibly originally continuous variables have been discretized. One reason for focusing on discrete data is that in this case we can model the domain variables by multinomial distributions without having to make restricting assumptions about unimodality, normality etc., which is the situation we face in the continuous case.

A *clustering* of the data set $\mathbf{x}^n$ is here defined as a partitioning of the data into mutually exclusive subsets, the union of which forms the data set. The number of subsets is a priori unknown. The *clustering problem* is the task to determine the number of subsets, and to decide to which cluster each data vector belongs.

Formally, we can notate a clustering by using a *clustering vector* $\mathbf{z}^n = (z_1, \ldots, z_n)$, where z_j denotes the index of the cluster to which the data vector $\mathbf{x}_j$ is assigned to. Denote the *clustering variable* by Z so that $\mathbf{z}^n$ is a sample from the distribution of Z. The number of clusters, say K_0, is implicitly defined in the clustering vector, as it can be determined by counting the number of different values appearing in $\mathbf{z}^n$. It is reasonable to assume that K_0 is bounded by the size of our data set, so we can define the *clustering space* $\mathcal{Z}$ as the set containing all the clusterings $\mathbf{z}^n$ with the number of clusters being less or equal to n. Hence the clustering problem is now to find from all the $\mathbf{z}^n \in \mathcal{Z}$ the optimal clustering $\mathbf{z}^n$.

For solving the clustering problem we obviously need a global optimization criterion that can be used for comparing clusterings with different number of clusters. To formalize this, we first need to explicate the type of probabilistic models we consider.

As in [8], we use the finite mixture model family here. The corresponding model class with K_0 components is denoted by $\mathcal{M}(K_0)$ and

$$\mathcal{M}(K_0) = \{P_{\text{FM}}(\cdot \mid \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta_{K_0}\}. \tag{9}$$

The basic finite mixture assumption is that given the value of the clustering variable Z, the primary variables $(X_1, \ldots, X_m)$ are independent. Consequently, we have

$$\begin{aligned} &P_{\text{FM}}(Z = z, X_1 = x_1, \ldots, X_m = x_m \mid \boldsymbol{\theta}) \\ &= P(Z = z \mid \boldsymbol{\theta}) \cdot \prod_{i=1}^{m} P(X_i = x_i \mid Z = z, \boldsymbol{\theta}). \end{aligned} \tag{10}$$

Furthermore, we assume that the distribution of $P(Z \mid \boldsymbol{\theta})$ is multinomial with parameters $(\pi_1, \ldots, \pi_{K_0})$, and each $P(X_i \mid Z = k, \boldsymbol{\theta})$ is multinomial with parameters $(\sigma_{ik1}, \ldots, \sigma_{ikK_i})$. The whole parameter space is then

$$\begin{aligned} \Theta_{K_0} = \{&(\pi_1, \ldots, \pi_{K_0}), \\ &(\sigma_{111}, \ldots, \sigma_{11K_1}), \ldots, (\sigma_{mK_01}, \ldots, \sigma_{mK_0K_m}) : \\ &\pi_k \geq 0,\ \sigma_{ikl} \geq 0,\ \pi_1 + \cdots + \pi_{K_0} = 1, \\ &\sigma_{ik1} + \cdots + \sigma_{ikK_i} = 1, \\ &i = 1, \ldots, m,\ k = 1, \ldots K_0\}, \end{aligned} \tag{11}$$

and the parameters are defined by $\pi_k = P(Z = k)$, $\sigma_{ikl} = P(X_i = l \mid Z = k)$.

Our optimality criterion for clustering is based on information-theoretical arguments, in particular on the Minimum Description Length (MDL) principle. Intuitively, the MDL principle aims at finding the shortest possible encoding for the data, in other words the goal is to find the most compressed representation of the data. Compression is possible by exploiting underlying regularities found in the data — the more regularities found, the higher the compression rate. Consequently, the MDL optimal encoding has found all the available regularities in the data; if there would be an "unused" regularity, this could be used for compressing the data even further.

What does this mean in the clustering framework? We suggest the following criterion for clustering: *the data vectors should be partitioned so that the vectors belonging to the same cluster can be compressed well together*. This means that those data vectors that obey the same set of underlying regularities are grouped together. In other words, the MDL clustering approach defines an implicit multilateral distance metric between the data vectors.

In [8], we suggested the following formalization of general optimality criterion for finding the optimal clustering $\hat{\mathbf{z}}^n$:

$$\hat{\mathbf{z}}^n = \arg\max_{\mathbf{z}^n} P(\mathbf{x}^n, \mathbf{z}^n \mid \mathcal{M}(K_0)). \tag{12}$$

From the coding point of view, definition (12) means the following: If one uses separate codes for encoding the data in different clusters, then in order to be able to decode the data, one needs to send with each vector the index of the corresponding code to be used. This means that we need to encode not only the data $\mathbf{x}^n$, but also the clustering $\mathbf{z}^n$, which is exactly what is done in (12).

There are, naturally, several ways to define the joint probability $P(\mathbf{x}^n, \mathbf{z}^n \mid \mathcal{M}(K_0))$. In [8], we compared the MDL and Bayesian approaches and the conclusion was that the MDL approach has several advantages over the Bayesian one. Firstly, the MDL principle does not assume that the chosen model class is correct. It even says that there is no such thing as a true model or model class, as acknowledged by many practitioners. The model class is only used as a technical device for constructing an efficient code. Secondly, there is no need to define a prior distribution for the parameters. The choice of the prior has a major effect on the quality of the results, as shown in [8]. Since there is no automatic way to choose the optimal prior, the Bayesian approach has a disadvantage here. Finally, the empirical results of [8] clearly favored the MDL approach, especially in the more complex cases. For these reasons, in the following we will only concentrate on the MDL approach.

As mentioned in Section 2, MDL model selection is based on the minimization of the stochastic complexity, which is the minus logarithm of the NML distribution. Assuming i.i.d., the NML distribution for the finite mixture model can be written as (see [8])

$$P_{\text{NML}}(\mathbf{x}^n, \mathbf{z}^n \mid \mathcal{M}(K_0)) = \frac{\prod_{k=1}^{K_0} \left(\frac{h_k}{n}\right)^{h_k} \prod_{i=1}^{m} \prod_{l=1}^{K_i} \left(\frac{f_{ikl}}{h_k}\right)^{f_{ikl}}}{\mathcal{C}_{\text{FM}}(\mathcal{M}(K_0), n)}, \quad (13)$$

where h_k is the number of times Z has value k in $\mathbf{z}^n$, f_{ikl} is the number of times X_i has value l when Z has value k, and $\mathcal{C}_{\text{FM}}(\mathcal{M}(K_0), n)$ is given by (see [8])

$$\mathcal{C}_{\text{FM}}(\mathcal{M}(K_0), n) = \sum_{h_1+\cdots+h_{K_0}=n} \frac{n!}{h_1! \cdots h_{K_0}!} \prod_{k=1}^{K_0} \left(\frac{h_k}{n}\right)^{h_k} \cdot \prod_{i=1}^{m} \mathcal{C}_{\text{MN}}(K_i, h_k), \quad (14)$$

and $\mathcal{C}_{\text{MN}}(K, n)$ is given by

$$\mathcal{C}_{\text{MN}}(K, n) = \sum_{h_1+\cdots+h_K=n} \frac{n!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k}. \quad (15)$$

The stochastic complexity for the finite mixture model can now be written as

$$SC(\mathbf{x}^n \mid \mathcal{M}(K_0)) = -\sum_{k=1}^{K_0} h_k \cdot \log \frac{h_k}{n} \sum_{i=1}^{m} \sum_{l=1}^{K_i} f_{ikl} \cdot \log \frac{f_{ikl}}{h_k} + \mathcal{C}_{\text{FM}}(\mathcal{M}(K_0), n), \quad (16)$$

While the terms $\mathcal{C}_{\text{MN}}(K, n)$ can be computed in linear time in n (see [25]), the sum (14) is clearly exponential and thus computationally infeasible. In [8], however, we presented an efficient recursive formula for computing this sum,

$$\mathcal{C}_{\text{FM}}(\mathcal{M}(K_0), n) = \sum_{r_1+r_2=n} \frac{n!}{r_1! r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \cdot \mathcal{C}_{\text{FM}}(\mathcal{M}(K_0^*), r_1) \cdot \mathcal{C}_{\text{FM}}(\mathcal{M}(K_0 - K_0^*), r_2), \quad (17)$$

where $1 \leq K_0^* \leq K_0 - 1$. A straightforward quadratic-time algorithm based on this formula presented in [8] allows the use of NML for practical clustering problems.

The clustering space $\mathcal{Z}$, however, is obviously exponential in size, which means that in practice we need to resort to combinatorial search algorithms in our attempt to solve the clustering problem. The search algorithm used in the empirical tests in [8] was a simple stochastic greedy algorithm. In the next section, we will compare five different algorithms for finding good clusterings using several real-world datasets from the UCI repository.

4 Empirical results

In this section, we will present two sets of results. The first set concentrates on finding the number of clusters and the actual clustering minimizing the stochastic complexity (16). In the second set of experiments, we will test how long it takes for each of the five algorithms to find the minimum SC value.

The first search algorithm candidate is a simple stochastic greedy (SG) algorithm, which was suggested in our previous paper [8]. The details of SG are described in Algorithm 1.

Algorithm 1 The stochastic greedy algorithm.

```
Choose a random initial clustering
repeat
    Choose a random data vector
    Move the chosen data vector to the cluster locally optimizing the SC score
until converged
```

Since our definition of clustering is based on the finite mixture model, the standard mixture learning algorithm,

EM (Expectation-Maximization) is a natural choice as a clustering search algorithm. The EM algorithm is iterative and consists of two alternating steps. In the E-step, the current parameters of the mixture model are used to fractionally assign each data vector to the clusters. In the M-step, the parameters are updated based on the fractional assignments. See [26, 27] for more details on the EM algorithm. To obtain an actual clustering from the fractional assignments, in this work the most probable cluster for each data vector is chosen after the EM algorithm has converged.

Our third candidate algorithm is the K-means algorithm (KM), sometimes called the CEM algorithm [28], is a simple modification to the EM algorithm. The difference is that in the E-step, each data vector is fully assigned to the most probable cluster, i.e., no fractional assignments are used.

Each of the described algorithms needs to be initialized prior to the iterative updating procedure. In our tests, we started each algorithm simply by choosing a random clustering. To test the importance of the initialization, we added two hybrid methods to our set of candidate search algorithms. The first hybrid algorithm (KMSG) starts by running the K-means algorithm until convergence and then switches to the stochastic greedy search. The second algorithm (EMSG) is the same except that the EM algorithm is used as an initializer.

It should be noted that we also tested several purely greedy algorithms, such as bottom-up and top-down clustering. However, we noticed very early that these algorithms are very slow and converge to highly suboptimal local optima and consequently were dropped from our tests.

Having fixed the set of candidate search algorithms, the next task is to define a strategy for finding the optimal number of clusters and the actual clustering. Since all the five algorithms converge to a local optimum of the stochastic complexity, the natural strategy is to restart the algorithms several times from different starting points.

Although the NML scoring criterion can be used for comparing clusterings with different number of clusters, the framework does not offer an explicit way to directly infer the optimal number of clusters (K). Consequently, the second part of our search strategy is to vary the parameter K. The complete search strategy is described in Algorithm 2.

In the first batch of results we tested which of the five algorithms finds the best clusterings in terms of the stochastic complexity. Description of the datasets and the results can be found in Figure 1. For all the five algorithms, the minimum SC value found and the corresponding number of clusters is recorded. For each dataset, the minimum stochastic complexity over the algorithms is in boldface.

Algorithm 2 The search strategy used in our tests.

```
repeat
  for all D in datasets do
    for K = 1 to 20 do
      Choose a random initial K-clustering for dataset D
      for all A in {SG, KM, EM, KMSG, EMSG} do
        Run the algorithm A until converged
      end for
    end for
  end for
until 50 restarts have been made
```

The first thing to notice about the results is that all the five algorithms seem to end up choosing a similar number of clusters. This means that all the algorithms are useful in the task of choosing the optimal number of clusters with respect to the stochastic complexity. However, when we look at the actual SC values, there are significant differences between the algorithms. Since SC can be interpreted as a quality of a clustering, these differences are important. The SG algorithm and the hybrid EMSG are clearly the best ones. One interesting observation is that EMSG beats SG clearly in some of the more complex cases, i.e., when the size of data and the optimal number of clusters is bigger, while EMSG is never significantly worse than SG. The KMSG algorithm is also reasonable good, but it is practically always worse than EMSG.

The traditional KM and EM algorithms are the worst of the candidate algorithms. Especially KM is in some cases extremely poor, which is alarming since KM is one of the most frequently used clustering algorithms. Furthermore, the EM algorithm beats KM every time, which suggests that it is easier to find good quality clusterings by exploiting the "soft clustering" space than by working in the "hard clustering" space alone. This observation was also made in [29].

In the second set of experiments we recorded how much CPU time (in seconds) each algorithm required for finding their respective optimal clustering. The results can be found in Figure 2. For these experiments, we used otherwise the same search strategy as before except that the number of restarts was only 10 and the results were averaged over 5 runs of Algorithm 2.

The most important thing to notice from these results is that the hybrid EMSG algorithm, which we above found to produce comparable or better results than SG, is almost always significantly faster than the SG algorithm proving the intuitive argument that choosing a good initial clustering is important. This makes the EMSG algorithm a clear overall winner in our experiments. The KMSG algorithm is also faster than SG, but slower than

dataset	size	#attrs	SG		KM		EM		KMSG		EMSG	
			K	SC	K	SC	K	SC	K	SC	K	SC
Australian	690	15	2	5834.5	2	5884.6	2	5844.5	2	**5833.8**	2	5834.5
Balance	625	5	2	**3795.0**	3	3811.5	2	3800.5	3	3809.3	2	**3795.0**
Dermatology	366	35	6	**8556.0**	5	9083.7	5	8792.0	6	**8556.0**	6	**8556.0**
Diabetes	768	9	4	**5137.7**	3	5245.9	3	5182.5	3	5158.0	5	5144.3
Ecoli	336	8	4	**2088.8**	3	2116.4	3	2090.9	3	2089.0	3	2089.0
Hepatitis	155	20	3	**2266.9**	3	2294.0	3	2287.8	3	**2266.9**	3	**2266.9**
Ionosphere	351	35	15	**10011.3**	13	10970.6	12	10339.8	17	10013.0	15	10012.7
Iris	150	5	4	**632.6**	3	634.6	4	633.5	3	633.9	4	**632.6**
Liver	345	7	2	**1689.6**	3	1727.4	2	1702.0	3	1702.9	2	**1689.6**
Lymphography	148	19	5	**2057.3**	5	2094.8	5	2074.5	5	**2057.3**	5	**2057.3**
Vehicle	846	19	13	10722.2	11	11227.0	13	10781.6	13	10712.8	13	**10710.0**
Tic-Tac-Toe	958	10	18	8921.5	17	9291.0	17	**8888.4**	19	8939.4	17	**8888.4**
Wine	178	14	3	**2402.2**	3	2440.8	3	2403.7	3	**2402.2**	3	**2402.2**
Yeast	1484	9	5	9338.3	6	9543.1	4	9385.3	5	9383.0	4	**9327.6**

Figure 1: The minimum SC scores and the number of clusters chosen by the candidate algorithms for the UCI datasets.

dataset	SG	KM	EM	KMSG	EMSG
Australian	1.0	0.3	0.2	0.4	0.3
Balance	1.3	3.4	0.2	0.5	0.2
Dermatology	11.0	14.4	19.7	8.2	7.4
Diabetes	2.0	14.3	5.4	3.7	2.7
Ecoli	0.6	3.2	3.3	0.9	0.2
Hepatitis	0.8	2.2	1.5	0.6	0.5
Ionosphere	121.2	11.8	7.2	132.7	103.5
Iris	0.3	0.4	0.3	0.3	0.2
Liver	0.4	2.5	0.2	0.3	0.2
Lymphography	1.2	2.2	2.1	0.6	0.8
Vehicle	52.6	17.7	48.1	59.9	59.6
Tic-Tac-Toe	1428.5	14.0	30.0	1217.4	240.3
Wine	0.3	1.8	4.4	0.3	0.3
Yeast	19.5	25.4	0.9	15.6	3.5

Figure 2: The CPU times (in seconds) spend by the candidate algorithms in finding the optimum clusterings.

EMSG. It is also noteworthy that KM and EM are often much slower than the other algorithms even though they produce inferior results. This makes the applicability of KM and EM even more questionable in the setting used here.

5 Conclusions

In this paper, we have extended our previously suggested framework for data clustering based on the idea that a good clustering is such that it allows efficient compression when the data are encoded together with the cluster labels. As a first extension we introduced five optimization algorithms for minimizing the stochastic complexity. Secondly, using these algorithms, we conducted an extensive set of experiments with several real-world datasets. In the first part of the tests we recorded the number of clusters chosen and the quality of the actual clusterings found by the algorithms. The idea of the second batch of tests was to see how much CPU time each algorithm requires for finding the best solution.

In the empirical results we found out that all the five algorithms were useful if the goal is to find the NML optimal number of clusters. However, the quality of the individual clusterings found by the more traditional KM and EM algorithms was questionable. These algorithms were also found to be slow. The most interesting observation was that the novel hybrid EMSG algorithm produced the best results and was also significantly faster than the SG algorithm used in our previous work.

In these tests, our search strategy was a very simple one. It is a natural topic of our future research to test more elaborate strategies, such as trying to find the optimal number of clusters in a more efficient way than what we did here. Another interesting extension is the development of stochastic greedy type of SC optimization algorithms that would be capable of exploiting the soft clustering search space in a similar manner EM does.

Acknowledgment

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.

References

[1] A.K. Jain, M.N. Murty, and P.J Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[2] P. Smyth. Probabilistic model-based clustering of multivariate and sequential data. In D. Heckerman and J. Whittaker, editors, *Proceedings of the Seventh International Conference on Artificial Intelligence and Statistics*, pages 299–304. Morgan Kaufmann Publishers, 1999.

[3] Chris Fraley and Adrian E. Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[4] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman. Autoclass: A Bayesian classification system. In *Proceedings of the Fifth International Conference on Machine Learning*, pages 54–64, Ann Arbor, June 1988.

[5] B.S. Everitt and D.J. Hand. *Finite Mixture Distributions*. Chapman and Hall, London, 1981.

[6] D.M. Titterington, A.F.M. Smith, and U.E. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, New York, 1985.

[7] G.J. McLachlan, editor. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

[8] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri. An MDL framework for data clustering. In P. Grünwald, I.J. Myung, and M. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*. The MIT Press, 2006.

[9] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:445–471, 1978.

[10] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society*, 49(3):223–239 and 252–265, 1987.

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

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

[13] A. Barron, J. Rissanen, and B. Yu. The minimum description principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, October 1998.

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

[15] J. Rissanen. Hypothesis selection and testing by the MDL principle. *Computer Journal*, 42(4):260–269, 1999.

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

[17] 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, July 2001.

[18] B. Dom. An information-theoretic external cluster-validity measure. Technical Report RJ 10219, IBM Research, 2001.

[19] M. Plumbley. Clustering of sparse binary data using a minimum description length approach. Technical report, Department of Electrical Engineering, Queen Mary, University of London, 2002. Unpublished manuscript.

[20] M-C. Ludl and G. Widmer. Clustering criterion based on minimum length encoding. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Proceedings of the 13th European Conference on Machine Learning*, volume 2430 of *Lecture Notes in Computer Science*, pages 258–269. Springer, 2002.

[21] S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases, University of California, Irvine, Dept. of Information and Computer Sciences, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[22] P. Grünwald. Minimum description length tutorial. In P. Grünwald, I.J. Myung, and M. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*, pages 23–79. The MIT Press, 2006.

[23] J. Rissanen. *Information and Complexity in Statistical Modeling*. Springer, 2007.

[24] V. Balasubramanian. MDL, Bayesian inference, and the geometry of the space of probability distributions. In P. Grünwald, I.J. Myung, and M. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*, pages 81–98. The MIT Press, 2006.

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

[26] 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*, 39(1):1–38, 1977.

[27] P. Kontkanen, P. Myllymäki, and H. Tirri. Constructing Bayesian finite mixture models by the EM algorithm. Technical Report NC-TR-97-003, ESPRIT Working Group on Neural and Computational Learning (NeuroCOLT), 1996.

[28] M. Meila and D. Heckerman. An experimental comparison of several clustering and initialization methods. In Gregory F. Cooper and Serafín Moral, editors, *UAI'98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 386–395, 1998.

[29] G. Hamerly and C. Elkan. Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM'02)*, pages 600–607, November 2002.

TIETOJENKÄSITTELYTIETEEN LAITOS
PL 68 (Gustaf Hällströmin katu 2 b)
00014 Helsingin yliopisto

DEPARTMENT OF COMPUTER SCIENCE
P.O. Box 68 (Gustaf Hällströmin katu 2 b)
FIN-00014 University of Helsinki, FINLAND

JULKAISUSARJA **A**

SERIES OF PUBLICATIONS **A**

Reports may be ordered from: Kumpula Science Library, P.O. Box 64, FIN-00014 University of Helsinki, FINLAND.

A-2003-1 J. Lindström: Optimistic concurrency control methods for real-time database systems. 111 pp. (Ph.D. Thesis)

A-2003-2 H. Helin: Supporting nomadic agent-based applications in the FIPA agent architecture. 200+17 pp. (Ph.D. Thesis)

A-2003-3 S. Campadello: Middleware infrastructure for distributed mobile applications. 164 pp. (Ph.D. Thesis)

A-2003-4 J. Taina: Design and analysis of a distributed database architecture for IN/GSM data. 130 pp. (Ph.D. Thesis)

A-2003-5 J. Kurhila: Considering individual differences in computer-supported special and elementary education. 135 pp. (Ph.D. Thesis)

A-2003-6 V. Mäkinen: Parameterized approximate string matching and local-similarity-based point-pattern matching. 144 pp. (Ph.D. Thesis)

A-2003-7 M. Luukkainen: A process algebraic reduction strategy for automata theoretic verification of untimed and timed concurrent systems. 141 pp. (Ph.D. Thesis)

A-2003-8 J. Manner: Provision of quality of service in IP-based mobile access networks. 191 pp. (Ph.D. Thesis)

A-2004-1 M. Koivisto: Sum-product algorithms for the analysis of genetic risks. 155 pp. (Ph.D. Thesis)

A-2004-2 A. Gurtov: Efficient data transport in wireless overlay networks. 141 pp. (Ph.D. Thesis)

A-2004-3 K. Vasko: Computational methods and models for paleoecology. 176 pp. (Ph.D. Thesis)

A-2004-4 P. Sevon: Algorithms for Association-Based Gene Mapping. 101 pp. (Ph.D. Thesis)

A-2004-5 J. Viljamaa: Applying Formal Concept Analysis to Extract Framework Reuse Interface Specifications from Source Code. 206 pp. (Ph.D. Thesis)

A-2004-6 J. Ravantti: Computational Methods for Reconstructing Macromolecular Complexes from Cryo-Electron Microscopy Images. 100 pp. (Ph.D. Thesis)

A-2004-7 M. Kääriäinen: Learning Small Trees and Graphs that Generalize. 45+49 pp. (Ph.D. Thesis)

A-2004-8 T. Kivioja: Computational Tools for a Novel Transcriptional Profiling Method. 98 pp. (Ph.D. Thesis)

A-2004-9 H. Tamm: On Minimality and Size Reduction of One-Tape and Multitape Finite Automata. 80 pp. (Ph.D. Thesis)

A-2005-1 T. Mielikäinen: Summarization Techniques for Pattern Collections in Data Mining. 201 pp. (Ph.D. Thesis)

A-2005-2 A. Doucet: Advanced Document Description, a Sequential Approach. 161 pp. (Ph.D. Thesis)

A-2006-1 A. Viljamaa: Specifying Reuse Interfaces for Task-Oriented Framework Specialization. 285 pp. (Ph.D. Thesis)

A-2006-2 S. Tarkoma: Efficient Content-based Routing, Mobility-aware Topologies, and Temporal Subspace Matching. 198 pp. (Ph.D. Thesis)

A-2006-3 M. Lehtonen: Indexing Heterogeneous XML for Full-Text Search. 185+3 pp. (Ph.D. Thesis)

A-2006-4 A. Rantanen: Algorithms for ^{13}C Metabolic Flux Analysis. 92+73 pp. (Ph.D. Thesis)

A-2006-5 E. Terzi: Problems and Algorithms for Sequence Segmentations. 141 pp. (Ph.D. Thesis)

A-2007-1 P. Sarolahti: TCP Performance in Heterogeneous Wireless Networks. (Ph.D. Thesis)

A-2007-2 M. Raento: Exploring privacy for ubiquitous computing: Tools, methods and experiments. (Ph.D. Thesis)

A-2007-3 L. Aunimo: Methods for Answer Extraction in Textual Question Answering. 127+18 pp. (Ph.D. Thesis)

A-2007-4 T. Roos: Statistical and Information-Theoretic Methods for Data Analysis. 82+75 pp. (Ph.D. Thesis)

A-2007-5 S. Leggio: A Decentralized Session Management Framework for Heterogeneous Ad-Hoc and Fixed Networks. 230 pp. (Ph.D. Thesis)

A-2007-6 O. Riva: Middleware for Mobile Sensing Applications in Urban Environments. 195 pp. (Ph.D. Thesis)

A-2007-7 K. Palin: Computational Methods for Locating and Analyzing Conserved Gene Regulatory DNA Elements. 130 pp. (Ph.D. Thesis)

A-2008-1 I. Autio: Modeling Efficient Classification as a Process of Confidence Assessment and Delegation. 212 pp. (Ph.D. Thesis)

A-2008-2 J. Kangasharju: XML Messaging for Mobile Devices. 24+255 pp. (Ph.D. Thesis).

A-2008-3 N. Haiminen: Mining Sequential Data – in Search of Segmental Structures. 60+78 pp. (Ph.D. Thesis)

A-2008-4 J. Korhonen: IP Mobility in Wireless Operator Networks. (Ph.D. Thesis)

A-2008-5 J.T. Lindgren: Learning nonlinear visual processing from natural images. 100+64 pp. (Ph.D. Thesis)

A-2009-1 K. Hätönen: Data mining for telecommunications network log analysis. 153 pp. (Ph.D. Thesis)

A-2009-2 T. Silander: The Most Probable Bayesian Network and Beyond. (Ph.D. Thesis)

A-2009-3 K. Laasonen: Mining Cell Transition Data. 148 pp. (Ph.D. Thesis)

A-2009-4 P. Miettinen: Matrix Decomposition Methods for Data Mining: Computational Complexity and Algorithms. 164+6 pp. (Ph.D. Thesis)

A-2009-5 J. Suomela: Optimisation Problems in Wireless Sensor Networks: Local Algorithms and Local Graphs. 106+96 pp. (Ph.D. Thesis)

A-2009-6 U. Köster: A Probabilistic Approach to the Primary Visual Cortex. 168 pp. (Ph.D. Thesis)

A-2009-7 P. Nurmi: Identifying Meaningful Places. 83 pp. (Ph.D. Thesis)

A-2009-8 J. Makkonen: Semantic Classes in Topic Detection and Tracking. 155 pp. (Ph.D. Thesis)

A-2009-9 P. Rastas: Computational Techniques for Haplotype Inference and for Local Alignment Significance. 64+50 pp. (Ph.D. Thesis)

A-2009-10 T. Mononen: Computing the Stochastic Complexity of Simple Probabilistic Graphical Models. 59+46 pp. (Ph.D. Thesis)