HELSINKI
INSTITUTE FOR
INFORMATION
TECHNOLOGY

# AN MDL FRAMEWORK FOR DATA CLUSTERING

Petri Kontkanen, Petri Myllymäki, Wray Buntine,
Jorma Rissanen and Henry Tirri

October 15, 2003

HIIT
TECHNICAL
REPORT
2004–6

# AN MDL FRAMEWORK FOR DATA CLUSTERING

Petri Kontkanen, Petri Myllymäki, Wray Buntine, Jorma Rissanen and Henry Tirri

Helsinki Institute for Information Technology HIIT
Tammasaarenkatu 3, Helsinki, Finland
PO BOX 9800
FI-02015 TKK, Finland
`http://www.hiit.fi`

HIIT Technical Reports 2004–6
ISSN 1458-9478
URL: http://cosco.hiit.fi/Articles/hiit-2004-6.pdf

# 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 $\Omega$ 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 $\Omega$ 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 $\Theta$ 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 $\theta$ 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 $\theta$ 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:

$$\begin{aligned} SC(\mathbf{x}^n \mid M) &= -\log \frac{P(\mathbf{x}^n, \hat{\theta}(\mathbf{x}^n) \mid \theta, M)\Big|_{\theta=\hat{\theta}(\mathbf{x}^n)}}{R_M^n} \\ &= -\log P(\hat{\theta}(\mathbf{x}^n)|n, M) - \log P(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), \theta, M)\Big|_{\theta=\hat{\theta}(\mathbf{x}^n)} \qquad (1.11) \end{aligned}$$

where

$$P(\hat{\theta}(\mathbf{x}^n)|n, M) = \frac{P(\hat{\theta}(\mathbf{x}^n) \mid \theta, M)\Big|_{\theta=\hat{\theta}(\mathbf{x}^n)}}{\sum_{\hat{\theta}} P(\hat{\theta}(\mathbf{x}^n) = \hat{\theta} \mid \theta, M)\Big|_{\theta=\hat{\theta}(\mathbf{x}^n)}} . \qquad (1.12)$$

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

$$P(\hat{\theta}(\mathbf{x}^n)|n, M) \ \propto \ \max_{\theta} P(\hat{\theta}(\mathbf{x}^n) \mid \theta, M) \ . \qquad (1.13)$$

Thus the parameter space $\Theta$ has been discretized to values achieving a maximum likelihood for some sample of size $n$, and the prior distributed so each has its

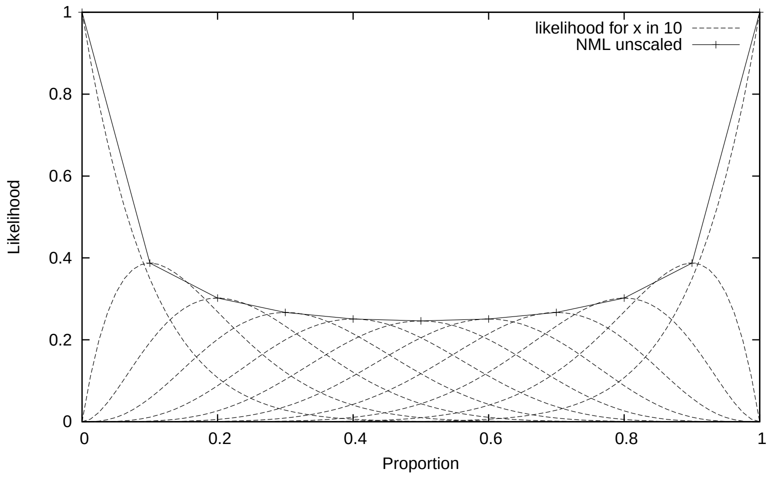

**Figure 1.1** Likelihood curves for K=2, n=10.

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 $\theta$ 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 $\Theta$, 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

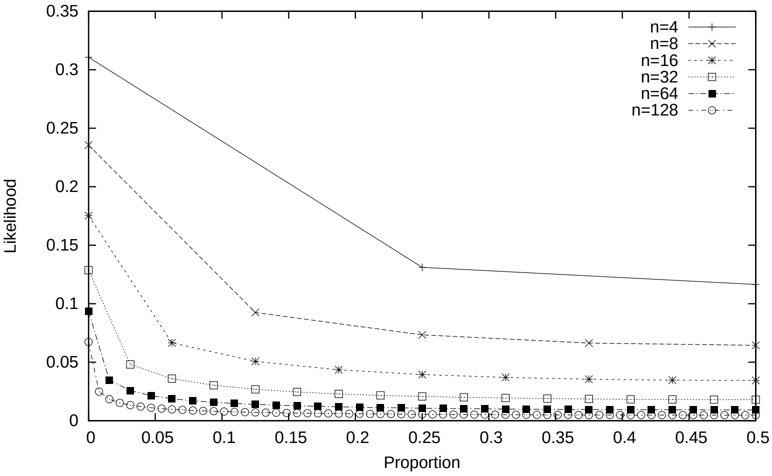

**Figure 1.2** NML distribution for K=2, different n.

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 $\theta$ 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 $\theta$: Find a $Q(\cdot)$

minimizing the worst case (for $\theta$) increase over the average best case codelength for $\theta$,

$$\begin{aligned}
&\min_{Q(\cdot)} \max_{\theta} E_{P(\mathbf{x}^n|\theta,M)} \log \frac{P(\mathbf{x}^n|\hat{\theta}(\mathbf{x}^n),M)}{Q(\mathbf{x}^n)} \\
&= \min_{Q(\cdot)} \max_{P(\theta|M)} E_{P(\theta|M)} E_{P(\mathbf{x}^n|\theta,M)} \log \frac{P(\mathbf{x}^n|\hat{\theta}(\mathbf{x}^n),M)}{Q(\mathbf{x}^n)} \\
&= \max_{P(\theta|M)} E_{P(\theta|M)} E_{P(\mathbf{x}^n|\theta,M)} \log \frac{P(\mathbf{x}^n|\hat{\theta}(\mathbf{x}^n),M)}{P(\mathbf{x}^n|M)} \\
&= \log R_M^n - \min_{P(\theta|M)} KL\left(P(\mathbf{x}^n|M) \| SC(\mathbf{x}^n|M)\right). \qquad (1.15)
\end{aligned}$$

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 $\theta$ is equivalent to finding a prior $P(\theta|M)$ maximizing the average over $\theta$, 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]

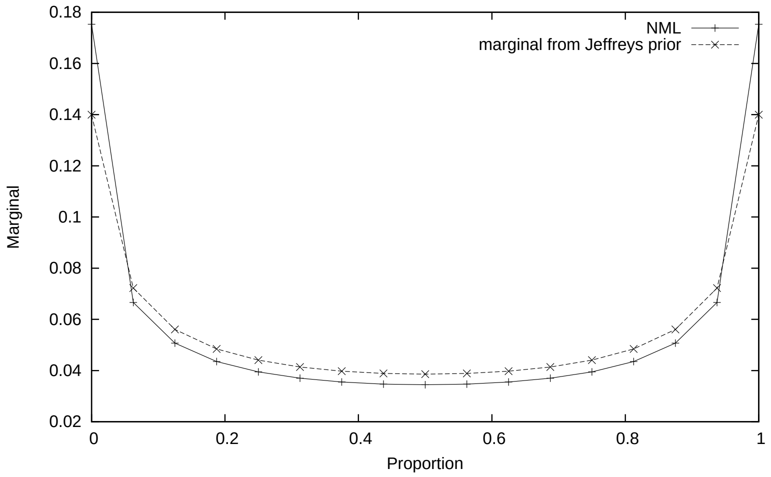

**Figure 1.3** Jeffreys prior versus NML as $P(\hat{\theta}|n = 16, M)$ for binomial.

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.

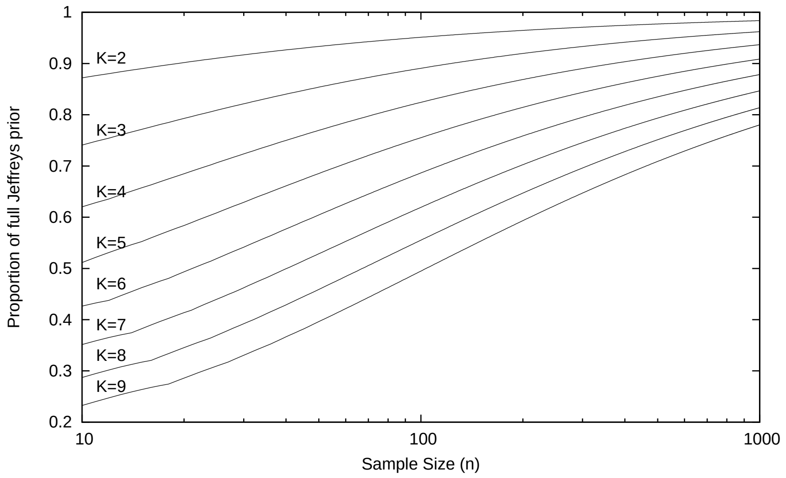

**Figure 1.4** Proportion of Jeffreys prior in NML for the multinomial model.

### *1.4.1.1 Multinomial maximum likelihood*

Let us assume that we have a multinomial variable $X$ with $K$ values. The parameter set $\Theta$ 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 $\theta_k$, 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

$$\begin{tikzcd}
{[R_2^j]} \arrow[rr] \arrow[drr] & & {[R_4^j]} \arrow[rr] & & {[R_8^j]} \arrow[rr] \arrow[dll] & & {[R_{16}^j]} \arrow[dll] \\
& & {[R_{10}^j]} \arrow[dr] & & {[R_{16}^j]} \arrow[dl] & & \\
& & & {[R_{26}^j]} & & &
\end{tikzcd}$$

**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

$$\begin{aligned}
R_K^n &= \sum_{h_1+\cdots+h_K=n} \frac{n!}{h_1!\cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k} = \sum_{h_1+\cdots+h_K=n} \frac{n!}{n^n} \prod_{k=1}^{K} \frac{h_k^{h_k}}{h_k!} \\
&= \sum_{\substack{h_1+\cdots+h_{K^*}=r_1\\ h_{K^*+1}+\cdots+h_K=r_2\\ r_1+r_2=n}} \frac{n!}{n^n} \frac{r_1^{r_1}}{r_1!} \frac{r_2^{r_2}}{r_2!} \left( \frac{r_1!}{r_1^{r_1}} \prod_{k=1}^{K^*} \frac{h_k^{h_k}}{h_k!} \cdot \frac{r_2!}{r_2^{r_2}} \prod_{k=K^*+1}^{K} \frac{h_k^{h_k}}{h_k!} \right) \\
&= \sum_{\substack{h_1+\cdots+h_{K^*}=r_1\\ h_{K^*+1}+\cdots+h_K=r_2\\ r_1+r_2=n}} \frac{n!}{n^n} \frac{r_1^{r_1}}{r_1!} \frac{r_2^{r_2}}{r_2!} \Bigg( \frac{r_1!}{h_1!\cdots h_{K^*}!} \prod_{k=1}^{K^*} \left(\frac{h_k}{r_1}\right)^{h_k} \\
&\qquad\qquad \cdot \frac{r_2!}{h_{K^*+1}!\cdots h_K!} \prod_{k=K^*+1}^{K} \left(\frac{h_k}{r_2}\right)^{h_k} \Bigg) \\
&= \sum_{r_1+r_2=n} \frac{n!}{r_1! r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \cdot R_{K^*}^{r_1} \cdot R_{K-K^*}^{r_2}.
\end{aligned} \tag{1.23}$$

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

$$
\begin{aligned}
\mathcal{R}^{n}_{\mathcal{M}_T,K} &= \sum_{h_1+\cdots+h_K=n} \left( \frac{n!}{h_1!\cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{n}\right)^{h_k} \prod_{i=1}^{m}\prod_{k=1}^{K} \mathcal{R}^{h_k}_{n_i} \right) \\
&= \sum_{h_1+\cdots+h_K=n} \left( \frac{n!}{n^n} \prod_{k=1}^{K} \frac{h_k^{h_k}}{h_k!} \prod_{i=1}^{m}\prod_{k=1}^{K} \mathcal{R}^{h_k}_{n_i} \right) \\
&= \sum_{\substack{h_1+\cdots+h_{K^*}=r_1 \\ h_{K^*+1}+\cdots+h_K=r_2 \\ r_1+r_2=n}} \left[ \frac{n!}{n^n}\,\frac{r_1^{r_1}}{r_1!}\,\frac{r_2^{r_2}}{r_2!} \left( \frac{r_1!}{r_1^{r_1}} \prod_{k=1}^{K^*} \frac{h_k^{h_k}}{h_k!} \cdot \frac{r_2!}{r_2^{r_2}} \prod_{k=K^*+1}^{K} \frac{h_k^{h_k}}{h_k!} \right) \right. \\
&\qquad\qquad \left. \cdot \prod_{i=1}^{m}\prod_{k=1}^{K^*} \mathcal{R}^{h_k}_{n_i} \prod_{k=K^*+1}^{K} \mathcal{R}^{h_k}_{n_i} \right], \qquad (1.27)
\end{aligned}
$$

from which we get the result

$$
\begin{aligned}
\mathcal{R}^{n}_{\mathcal{M}_T,K} &= \sum_{\substack{h_1+\cdots+h_{K^*}=r_1 \\ h_{K^*+1}+\cdots+h_K=r_2 \\ r_1+r_2=n}} \left[ \frac{n!}{r_1!r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \right. \\
&\qquad \cdot \left( \frac{r_1!}{h_1!\cdots h_{K^*}!} \prod_{k=1}^{K^*} \left(\frac{h_k}{r_1}\right)^{h_k} \prod_{i=1}^{m}\prod_{k=1}^{K^*} \mathcal{R}^{h_k}_{n_i} \right) \\
&\qquad \left. \cdot \left( \frac{r_2!}{h_{K^*+1}!\cdots h_K!} \prod_{k=K^*+1}^{K} \left(\frac{h_k}{r_2}\right)^{h_k} \prod_{i=1}^{m}\prod_{k=K^*+1}^{K} \mathcal{R}^{h_k}_{n_i} \right) \right] \\
&= \sum_{r_1+r_2=n} \frac{n!}{r_1!r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \cdot \mathcal{R}^{r_1}_{\mathcal{M}_T,K^*} \cdot \mathcal{R}^{r_2}_{\mathcal{M}_T,K-K^*}. \qquad (1.28)
\end{aligned}
$$

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.

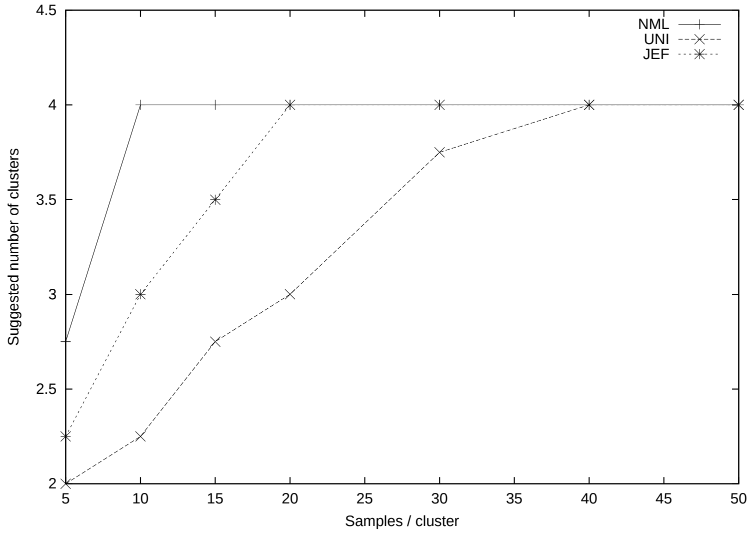

**Figure 1.6** An example of the behavior of different clustering scores in the task of finding a four cluster data partitioning, as a function of sample size per cluster.

In Figure 1.6 we see a typical example of how the NML, UNI and JEF clustering

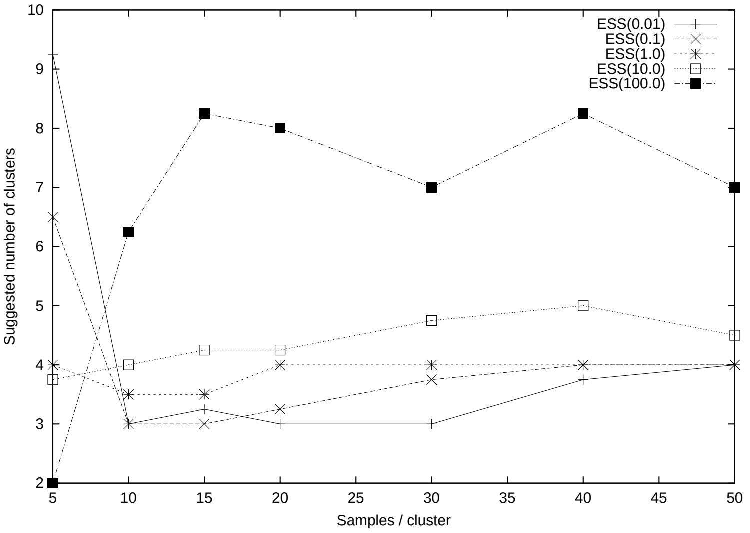

**Figure 1.7** An example of the behavior of different ESS clustering scores in the task of finding a four cluster data partitioning, as a function of sample size per cluster.

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.

# Index