Information-Theoretically Optimal Histogram Density Estimation

Petri Kontkanen, Petri Myllymäki

March 17, 2006

HIIT
TECHNICAL
REPORT
2006–2

Information-Theoretically Optimal Histogram Density Estimation

Petri Kontkanen, Petri Myllymäki

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

HIIT Technical Reports 2006–2
ISSN 1458-9478
URL: http://cosco.hiit.fi/Articles/hiit-2006-2.pdf

Information-Theoretically Optimal Histogram Density Estimation

Petri Kontkanen, Petri Myllymäki
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
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. MDL-based model selection is formalized via the normalized maximum likelihood (NML) distribution, which has several desirable optimality properties. We show how this approach can be applied for learning generic, irregular (variable-width bin) histograms, and how to compute the model selection criterion efficiently. We also derive a dynamic programming algorithm for finding both the NML-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 random sample of observations from an unknown density, 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 very 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 [15] 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 the cut point sets is then measured by some model selection criteria.

Our approach is based on information theory, more specifically on the *minimum encoding* or *minimum complexity* methods. These methods perform induction by seeking a theory that allows the most compact encoding of both the theory and available data. Intuitively speaking, 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, 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.

The most well-founded formalization of the minimum encoding approach is the *minimum description length* (MDL) principle developed by Rissanen [10, 11, 12]. 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.

The way MDL does model selection is 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 [17, 12]. The NML distribution has several theoretical optimality properties, which make it a very attractive candidate for performing model selection. It was originally [12, 2] formulated as a unique solution to the minimax problem presented in [17], which implied that NML is the minimax optimal universal model. Later [13], it was shown that NML is also the solution to a related problem involving expected regret. See Section 2 and [2, 13, 5, 14] 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 [9], 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 [16], 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 cases one must face severe computational problems with NML. The definition of the NML involves a normalizing integral or 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 criteria 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. [3, 6, 18], consider regular histograms only. Most similar to our work is [15], in which irregular histograms are learned with the Bayesian mixture criterion using a uniform prior. The same criterion is also used in [6], but the histograms are equal-width only. Another similarity between our work and [15] 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 MDL AND NML

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 several desirable properties. Firstly, it automatically protects against overfitting when learning both the parameters and the structure of the model. Secondly, there is no need to assume that there exists some underlying "true" model, which is not the case with other statistical methods. 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.

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

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

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

Denote the maximum likelihood estimate of data $\mathbf{x}^n$ by $\hat{\theta}(\mathbf{x}^n)$, i.e.,

$$\hat{\theta}(\mathbf{x}^n) = \arg\max_{\theta \in \Theta}\{f(\mathbf{x}^n \mid \theta)\}. \quad (2)$$

The *normalized maximum likelihood (NML)* density [17] is now defined as

$$f_{\mathrm{NML}}(\mathbf{x}^n \mid \mathcal{M}) = \frac{f(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), \mathcal{M})}{\mathcal{R}^n_{\mathcal{M}}}, \quad (3)$$

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

$$\mathcal{R}^n_{\mathcal{M}} = \int_{\mathbf{x}^n \in \mathcal{X}^n} f(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), \mathcal{M}) d\mathbf{x}^n, \quad (4)$$

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

$$SC(\mathbf{x}^n \mid \mathcal{M}) = -\log f_{\mathrm{NML}}(\mathbf{x}^n \mid \mathcal{M}) \quad (5)$$

$$= -\log f(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), \mathcal{M}) + \log \mathcal{R}^n_{\mathcal{M}}, \quad (6)$$

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 [1] for more discussion on this topic.

The NML density (3) has several important theoretical optimality properties. The first one is that NML provides a unique solution to the minimax problem posed in [17],

$$\min_{\hat{f}} \max_{\mathbf{x}^n} \log \frac{f(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n))}{\hat{f}(\mathbf{x}^n)} = \log \mathcal{R}^n_{\mathcal{M}}, \quad (7)$$

This means that the NML density is the *minimax optimal universal model*. A related property of NML involving expected regret was proven in [13]. This property states that NML also minimizes

$$\min_{\hat{f}} \max_{g} E_g \log \frac{f(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n))}{\hat{f}(\mathbf{x}^n)} = \log \mathcal{R}^n_{\mathcal{M}}, \quad (8)$$

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 $\mathbf{x}^n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ from an unknown density f on the interval $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$. Without any loss of generality, we assume that the data is sorted into increasing order. Typically, $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ are defined as the minimum and maximum value in $\mathbf{x}^n$, respectively. 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

$$\mathcal{X} = \{\mathbf{x}_{\min} + t\epsilon : t = 0, \ldots, \frac{\mathbf{x}_{\max} - \mathbf{x}_{\min}}{\epsilon}\}. \quad (9)$$

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):

$$([\mathbf{x}_{\min} - \epsilon/2, c_1],]c_1, c_2], \ldots,]c_{K-1}, \mathbf{x}_{\max} + \epsilon/2]). \quad (10)$$

The points c_k are called the *cut points* of the histogram. Note that in order to simplify the formulations, the original data range $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$ is extended by $\epsilon/2$ from both ends. 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

$$\mathcal{C} = \{\mathbf{x}_{\min} + \epsilon/2 + t\epsilon : t = 0, \ldots, \frac{\mathbf{x}_{\max} - \mathbf{x}_{\min}}{\epsilon} - 1\}, \quad (11)$$

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 $\theta \in \Theta$,

$$\Theta = \{(\theta_1, \ldots, \theta_K) : \theta_k \geq 0, \theta_1 + \cdots + \theta_K = 1\}, \quad (12)$$

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

$$f_h(x \mid \theta, C) = \frac{\epsilon \cdot \theta_k}{L_k}, \quad (13)$$

where $x \in]c_{k-1}, c_k]$. Note that (13) does not define a density in the pure sense, since $f_h(x \mid \theta, C)$

is actually the probability that x falls into the interval $]x-\epsilon/2, x+\epsilon/2]$. The density version of (13) would be straightforward to derive by letting $\epsilon \to 0$, but we prefer the pre-discretized version, since real data has a finite accuracy anyway.

Given (13), the likelihood of the whole data sample $\mathbf{x}^n$ is easy to write. We have

$$f_h(\mathbf{x}^n \mid \theta, C) = \prod_{k=1}^{K} \left(\frac{\epsilon \cdot \theta_k}{L_k} \right)^{h_k}, \tag{14}$$

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

To instantiate the NML distribution (3) 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 (4). It is well-known that the ML parameters are given by the relative frequencies

$$\hat{\theta}_k = \frac{h_k}{n}, \tag{15}$$

so that we have

$$f_h(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), C) = \prod_{k=1}^{K} \left(\frac{\epsilon \cdot h_k}{L_k \cdot n} \right)^{h_k}. \tag{16}$$

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 (4) is replaced by a sum over the space $\mathcal{X}^n$. We have

$$\mathcal{R}^n_{h_K} = \sum_{\mathbf{x}^n \in \mathcal{X}^n} \prod_{k=1}^{K} \left(\frac{\epsilon \cdot h_k}{L_k \cdot n} \right)^{h_k} \tag{17}$$

$$= \sum_{h_1 + \cdots + h_K = n} \frac{n!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{L_k}{\epsilon} \right)^{h_k} \cdot \prod_{k=1}^{K} \left(\frac{\epsilon \cdot h_k}{L_k \cdot n} \right)^{h_k} \tag{18}$$

$$= \sum_{h_1 + \cdots + h_K = n} \frac{n!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{n} \right)^{h_k}, \tag{19}$$

where the term $(L_k/\epsilon)^{h_k}$ in (18) 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 (19) of the parametric complexity is still an exponential sum, we can compute it efficiently. It turns out that (19) is exactly the same as the parametric complexity of a K-valued multinomial, which we studied in [8]. In this work, we showed that the generating function of the infinite sequence $(\mathcal{R}^0_{h_K}, \mathcal{R}^1_{h_K}, \mathcal{R}^2_{h_K}, \ldots)$ is similar to the generating function of the so-called tree polynomials $t_n(y)$ [7] defined by

$$\frac{1}{(1 - T(z))^y} = \sum_{n \geq 0} t_n(y) \frac{z^n}{n!}, \tag{20}$$

where $T(z)$ is the *Cayley's tree function* [4, 7]. From a tree polynomial recursion presented in [7], we derived

$$\mathcal{R}^n_{h_K} = \mathcal{R}^n_{h_{K-1}} + \frac{n}{K-2} \mathcal{R}^n_{h_{K-2}}, \tag{21}$$

which holds for $K > 2$.

It is now straightforward to write a linear-time algorithm based on (21). The computation starts with the trivial case $\mathcal{R}^n_{h_1} \equiv 1$. The case $K = 2$ is a simple sum

$$\mathcal{R}^n_{h_2} = \sum_{h_1 + h_2 = n} \frac{n!}{h_1! h_2!} \left(\frac{h_1}{n} \right)^{h_1} \left(\frac{h_2}{n} \right)^{h_2}, \tag{22}$$

which clearly can be computed in time $\mathcal{O}(n)$. Finally, recursion (21) 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 (6) for the histogram model. We have

$$SC(\mathbf{x}^n \mid C)$$

$$= -\log \frac{f_h(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), C)}{\mathcal{R}^n_{h_K}} \tag{23}$$

$$= -\log \frac{\prod_{k=1}^{K} \left(\frac{\epsilon \cdot h_k}{L_k \cdot n} \right)^{h_k}}{\mathcal{R}^n_{h_K}} \tag{24}$$

$$= \sum_{k=1}^{K} -h_k (\log(\epsilon \cdot h_k) - \log(L_k \cdot n)) + \log \mathcal{R}^n_{h_K}. \tag{25}$$

Equation (25) is the basis for measuring the quality of NML histograms, i.e., comparing different cut point sets. In the next section we will discuss how NML-optimal histograms can be found in practice.

4 LEARNING 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

$$\begin{aligned}\tilde{\mathcal{C}} =&(\{\mathbf{x}_j - \epsilon/2 : \mathbf{x}_j \in \mathbf{x}^n\} \cup \{\mathbf{x}_j + \epsilon/2 : \mathbf{x}_j \in \mathbf{x}^n\}) \\ &\setminus \{\mathbf{x}_{\min} - \epsilon/2, \mathbf{x}_{\max} + \epsilon/2\}. \end{aligned} \tag{26}$$

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 (25), 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 [5].

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

$$\begin{aligned} &BSC(\mathbf{x}^n \mid E, K, C) \\ &\quad = SC(\mathbf{x}^n \mid C) + \log \binom{E}{K-1} \qquad (27) \\ &\quad = \sum_{k=1}^{K} -h_k \left(\log(\epsilon \cdot h_k) - \log(L_k \cdot n)\right) \\ &\qquad + \log \mathcal{R}_{h_K}^n + \log \binom{E}{K-1}. \qquad (28) \end{aligned}$$

It is clear that there are an exponential number of possible cut point sets, and thus an exhaustive search to minimize (28) 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

$$\tilde{\mathcal{C}} = \{\tilde{c}_1, \ldots, \tilde{c}_E\},\ \tilde{c}_1 < \tilde{c}_2 < \cdots < \tilde{c}_E. \tag{29}$$

We also define $\tilde{c}_{E+1} = \mathbf{x}_{\max} + \epsilon/2$. Denote

$$\hat{B}_{K,e} = \min_{C \subseteq \tilde{\mathcal{C}}} BSC(\mathbf{x}^{n_e} \mid E, K, C), \tag{30}$$

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 (28) 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 $BSC(\mathbf{x}^{n_{e_K}} \mid E, K, C)$ 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

$$\begin{aligned} &BSC(\mathbf{x}^{n_{e_K}} \mid E, K, C) \\ &\quad = BSC(\mathbf{x}^{n_{e_{K-1}}} \mid E, K-1, C') \\ &\qquad - (n_{e_K} - n_{e_{K-1}})(\log(\epsilon \cdot (n_{e_K} - n_{e_{K-1}})) \\ &\qquad - \log((\tilde{c}_{e_K} - \tilde{c}_{e_{K-1}}) \cdot n)) \\ &\qquad + \log \frac{\mathcal{R}_{h_K}^{n_{e_K}}}{\mathcal{R}_{h_{K-1}}^{n_{e_{K-1}}}} + \log \frac{E-K+2}{K-1}, \end{aligned} \tag{31}$$

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

$$\log \frac{\binom{E}{K-1}}{\binom{E}{K-2}} = \log \frac{E-K+2}{K-1}. \tag{32}$$

Similarly as in [15], we can now write the dynamic programming recursion as

$$\hat{B}_{K,e} = \min_{e'} \Bigg\{ \hat{B}_{K-1,e'} - (n_e - n_{e'}) \cdot (\log(\epsilon \cdot (n_e - n_{e'})) - \log((\tilde{c}_e - \tilde{c}_{e'}) \cdot n)) + \log \frac{\mathcal{R}^{n_e}_{h_K}}{\mathcal{R}^{n_{e'}}_{h_{K-1}}} + \log \frac{E-K+2}{K-1} \Bigg\}, \quad (33)$$

where $e' = K-1, \ldots, e-1$. The recursion is initialized with

$$\hat{B}_{1,e} = -n_e \cdot (\log(\epsilon \cdot n_e) - \log((\tilde{c}_e - (\mathbf{x}_{\min} - \epsilon/2)) \cdot n)), \quad (34)$$

for $e = 1, \ldots, E+1$. After that, the bin count is always increased by one, and (33) 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 (33) 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. 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 a fixed irregular 5-bin histogram (see below), and then used the dynamic programming method to find the MDL-optimal histograms. The accuracy parameter ϵ was fixed to 0.5. The sample sizes we used were 20, 50 and 200. The results can be found in Figures 1, 2 and 3, respectively. The top picture of each pair shows the optimized (minimum) value of the BSC criterion (28) as a function of the bin count K, and the bottom one visually compares the generating density (H_{gen}) with the MDL-optimal one (H_{nml}).

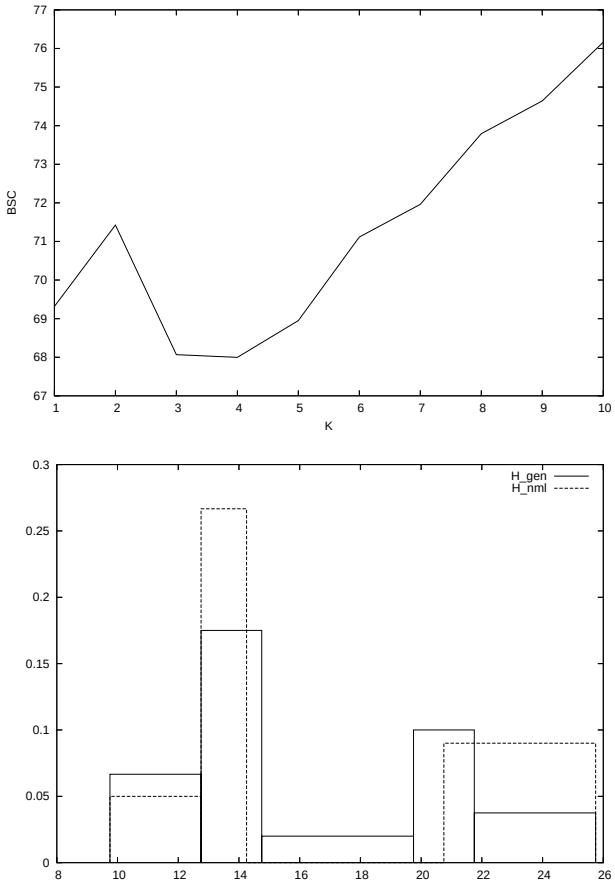

Figure 1: Simulation results with sample size 20. Above the optimal value of the BSC-criterion is plotted as a function of the bin count, below the generating density (solid line) is compared to the NML density (dashed line).

From the figures we can see that the NML histogram method clearly works well even with a small sample size. Although the bin count in Figure 1 is incorrectly estimated to be 4, the shape of the NML histogram is quite similar to the generating one, which is remarkable considering that the sample size is only 20. It is clear that any equal-width histogram estimator would be catastrophically bad in this situation. Note that the third bin of the NML histogram is empty. This is explained by the fact that the data sampling process did not produce any values in that interval, but that was not a problem for the NML method. Another interesting observation from Figure 1 is that the case $K = 1$ gets a better score than $K = 2$. The reason is that the parametric complexity and the codelength for encoding the cut points is higher in the latter case, and the increase in likelihood is not high enough to compensate that.

When the sample size is increased to 50 (Figure 2),

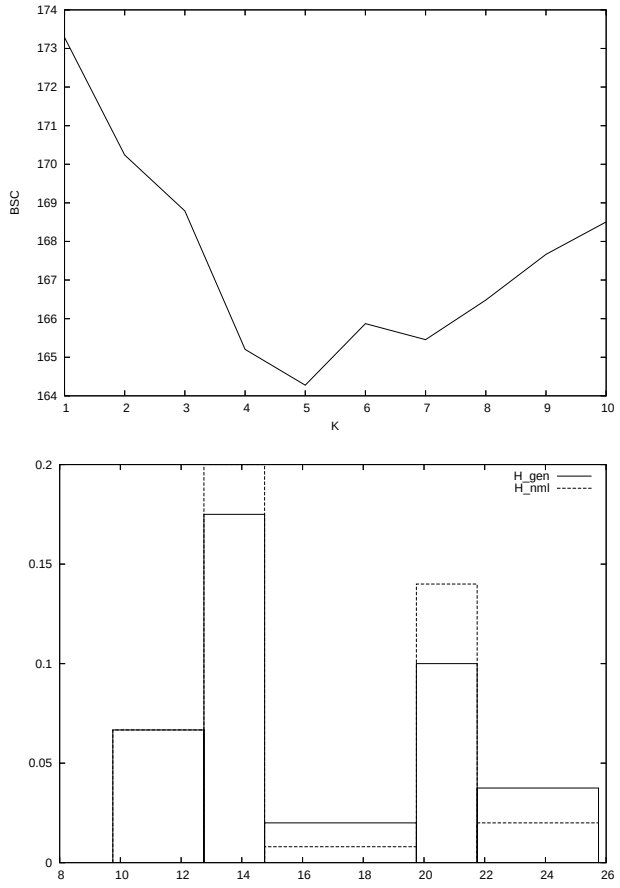

Figure 2: Simulation results with sample size 50. Above the optimal value of the BSC-criterion is plotted as a function of the bin count, below the generating density (solid line) is compared to the NML density (dashed line).

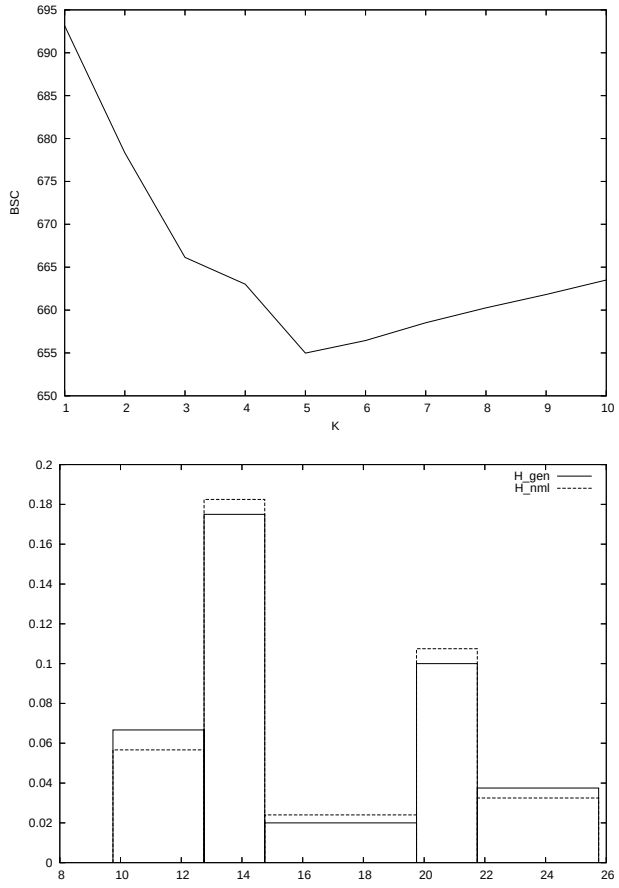

Figure 3: Simulation results with sample size 200. Above the optimal value of the BSC-criterion is plotted as a function of the bin count, below the generating density (solid line) is compared to the NML density (dashed line).

the bin count is correctly estimated to be 5. Furthermore, the cut points are exactly the same as with the generating density. The differences in bin heights are explained by the random nature of the data generating process. With sample size 200 (Figure 3), the NML-estimated histogram is practically the same as the generating one. The shape of the BSC-curve is also smoother than it was with smaller sample sizes.

6 CONCLUSION

In this paper we have presented an information-theoretic framework for histogram density estimation. The selected approach based on the MDL principle has several advantages. Firstly, the MDL criteria 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 criteria 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 criteria is based on the NML distribution, there is also no need to specify any prior distribution for the parameters.

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 was generated from a two-modal, ir-

regular histogram with 5 bins. From the tests it was clearly seen that the NML histogram method works well even when the sample size was only 20. When the sample size was increased, the quality of the estimated histogram got better, as expected.

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 [15], 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 Naive Bayes. These distributions are usually modeled with a normal distribution or a multinomial 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.

References

[1] 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, 2005.

[2] 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.

[3] 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.

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

[5] 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, 2005.

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

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

[8] P. Kontkanen and P. Myllymäki. Analyzing the stochastic complexity via tree polynomials. Technical Report 2005-4, Helsinki Institute for Information Technology (HIIT), 2005.

[9] 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.

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

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

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

[13] 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.

[14] J. Rissanen. Lectures on statistical modeling theory, August 2005. Available online at www.mdl-research.org.

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

[16] 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.

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

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