HELSINKI
INSTITUTE FOR
INFORMATION
TECHNOLOGY

COMPUTING THE REGRET TABLE FOR MULTINOMIAL DATA

Petri Kontkanen, Petri Myllymäki

April 12, 2005

HIIT
TECHNICAL
REPORT
2005–1

COMPUTING THE REGRET TABLE FOR MULTINOMIAL DATA

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 2005–1
ISSN 1458-9478
URL: http://cosco.hiit.fi/Articles/hiit-2005-1.pdf

NB. The HIIT Technical Reports series is intended for rapid dissemination of results produced by the HIIT researchers. Therefore, some of the results may also be later published as scientific articles elsewhere.

Computing the Regret Table for Multinomial Data

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

Abstract

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

1 Introduction

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

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

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

The most important notion of the MDL principle is the *Stochastic Complexity (SC)*, which is defined as the shortest description length of a given data relative to a model class $\mathcal{M}$. However, the applications of the modern, so called Normalized Maximum Likelihood (NML) version of SC, at least with multinomial data, have been quite rare. The modern definition of SC is based on the Normalized Maximum Likelihood (NML) code [Shtarkov, 1987]. Unfortunately, with multinomial data this code involves a sum over all the possible data matrices of certain length. Computing this sum, usually called the *regret*, is obviously exponential. Therefore, practical applications of the NML have been quite rare, In our previous work [Kontkanen *et al.*, 2003; 2005], we presented a polynomial time (quadratic) method to compute the regret. In this paper we improve our previous results and show how mathematical techniques such as discrete Fourier transform and fast convolution can be used in regret computation. The idea of applying these techniques to the regret computation problem was first suggested in [Koivisto, 2004], but as discussed in [Kontkanen *et al.*, 2005], in order to apply NML to practical tasks such as clustering, a whole table of regret terms is needed. We will modify the method of [Koivisto, 2004] for this task.

In Section 2 we shortly review how the stochastic complexity is defined and our previous work on the computational methods. We also introduce the concept of the regret table. Section 3 previews some mathematical results about convolution and discrete Fourier transform and presents the new fast convolution-based NML algorithm. Finally, Section 4 gives the concluding remarks and presents some ideas for future work.

2 Stochastic Complexity for Multinomial Data

2.1 NML And Stochastic Complexity

The most important notion of the MDL is the *Stochastic Complexity (SC)*. Intuitively, stochastic complexity is defined as the shortest description length of a given data relative to a model class. To formalize things, let us start with a definition of a model class. Consider a set $\Theta \in \mathbb{R}^d$, where d is a positive integer. A class of parametric distributions indexed by the elements of Θ is called a *model class*. That is, a model class $\mathcal{M}$ is defined as

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

Consider now a discrete data set (or matrix) $\mathbf{x}^N = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ of N outcomes, where each outcome $\mathbf{x}_j$ is an element of the set $\mathcal{X}$ consisting of all the vectors of the form $(a_1, \ldots, a_m)$, where each variable (or attribute) a_i takes on values $v \in \{1, \ldots, n_i\}$. Furthermore, let $\hat{\theta}(\mathbf{x}^N)$ denote the *maximum likelihood* estimate of data $\mathbf{x}^N$, i.e.,

$$\hat{\theta}(\mathbf{x}^N) = \arg\max_{\theta \in \Theta}\{P(\mathbf{x}^N \mid \theta)\}. \tag{2}$$

The *Normalized Maximum Likelihood (NML)* distribution [Shtarkov, 1987] is now defined as

$$P_{NML}(\mathbf{x}^N \mid \mathcal{M}) = \frac{P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M})}{\mathcal{R}^N_{\mathcal{M}}}, \tag{3}$$

where $\mathcal{R}^N_{\mathcal{M}}$ is given by

$$\mathcal{R}^N_{\mathcal{M}} = \sum_{\mathbf{x}^N} P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M}), \tag{4}$$

and the sum goes over all the possible data matrices of size N. The term $\mathcal{R}^N_{\mathcal{M}}$ is called the *regret*. The definition (3) is intuitively very appealing: every data matrix is modeled using its own maximum likelihood (i.e., best fit) model, and then a penalty for the complexity of the model class $\mathcal{M}$ is added to normalize the distribution.

The stochastic complexity of a data set $\mathbf{x}^N$ with respect to a model class $\mathcal{M}$ can now be defined as the negative logarithm of (3), i.e.,

$$SC(\mathbf{x}^n \mid \mathcal{M}) = -\log \frac{P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M})}{\mathcal{R}^N_{\mathcal{M}}} \tag{5}$$

$$= -\log P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M}) + \log \mathcal{R}^N_{\mathcal{M}}. \tag{6}$$

2.2 NML for Single Multinomial Model Class

In this section we instantiate the NML distribution (3) (and thus the stochastic complexity) for the single-dimensional multinomial case. Let us assume that a discrete random variable X with K values is multinomially distributed. That is, the parameter set Θ is a simplex

$$\Theta = \{(\theta_1, \ldots, \theta_K) : \theta_k \geq 0, \theta_1 + \cdots + \theta_K = 1\}, \tag{7}$$

where

$$\theta_k = P(X = k),\ k = 1, \ldots, K. \tag{8}$$

We denote this model class by $\mathcal{M}_K$. Now consider a data sample of size N from the distribution of X, i.e., $\mathbf{x}^N = (x_1, \ldots, x_N)$ and each $x_j \in \{1, \ldots, K\}$. The likelihood is clearly given by

$$P(\mathbf{x}^N \mid \theta) = \prod_{j=1}^N P(x_j \mid \theta) = \prod_{j=1}^N \theta_{x_j} = \prod_{k=1}^K \theta_k^{h_k}, \tag{9}$$

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 "sufficient" refers to the fact that the likelihood depends on the data only through them.

To instantiate the NML distribution (3) for the $\mathcal{M}_K$ model class, we need to find the maximum likelihood estimates for the parameters θ_k. As one might intuitively guess, the ML parameters are given by the relative frequencies of the values k in the data (see, e.g., [Johnson *et al.*, 1997]):

$$\hat{\theta}(\mathbf{x}^N) = (\hat{\theta}_1, \ldots, \hat{\theta}_K) \tag{10}$$

$$= (\frac{h_1}{N}, \ldots, \frac{h_K}{N}). \tag{11}$$

Thus, the likelihood evaluated at the maximum likelihood point is

$$P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N)) = \prod_{k=1}^K \left(\frac{h_k}{N}\right)^{h_k}, \tag{12}$$

and the NML distribution becomes

$$P_{NML}(\mathbf{x}^N \mid \mathcal{M}_K) = \frac{\prod_{k=1}^K \left(\frac{h_k}{N}\right)^{h_k}}{\mathcal{R}^N_{\mathcal{M}_K}}, \tag{13}$$

where

$$\mathcal{R}^N_{\mathcal{M}_K} = \sum_{\mathbf{x}^N} P(\mathbf{x}^N \mid \hat{\theta}(\mathbf{x}^N), \mathcal{M}_K) \tag{14}$$

$$= \sum_{h_1 + \cdots + h_K = N} \frac{N!}{h_1! \cdots h_K!} \prod_{k=1}^K \left(\frac{h_k}{N}\right)^{h_k}, \tag{15}$$

and the sum goes over all the *compositions* of N into K parts, i.e., over all the possible ways to choose a vector of non-negative integers $(h_1, \ldots, h_K)$ such that they sum up to N. The $N!/(h_1! \cdots h_K!)$ factor in (15) is called the *multinomial coefficient* and it is one of the basic combinatorial quantities. It counts the number of arrangements of N objects into K boxes each containing $h_1, \ldots, h_K$ objects, respectively.

An efficient method for computing (15) was derived in [Kontkanen *et al.*, 2003]. It is based on the following recursive formula:

$$\mathcal{R}^N_{\mathcal{M}_K} = \sum_{h_1 + \cdots + h_K = N} \frac{N!}{h_1! \cdots h_K!} \prod_{k=1}^K \left(\frac{h_k}{N}\right)^{h_k} \tag{16}$$

$$= \sum_{r_1 + r_2 = N} \frac{N!}{r_1! r_2!} \left(\frac{r_1}{N}\right)^{r_1} \left(\frac{r_2}{N}\right)^{r_2} \cdot \mathcal{R}^{r_1}_{\mathcal{M}_{k1}} \cdot \mathcal{R}^{r_2}_{\mathcal{M}_{k2}} \tag{17}$$

$$= \sum_{r=0}^N \frac{N!}{r!(N-r)!} \left(\frac{r}{N}\right)^r \left(\frac{N-r}{N}\right)^{N-r} \cdot \mathcal{R}^r_{\mathcal{M}_{k1}} \cdot \mathcal{R}^{N-r}_{\mathcal{M}_{k2}}, \tag{18}$$

where $k_1 + k_2 = K$. See [Kontkanen *et al.*, 2003] for details.

2.3 NML for Clustering Model Class

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

Let us assume that we have m variables, $(a_1, \ldots, a_m)$. We also assume the existence of a special variable c (which can be chosen to be one of the variables in our data or it can be latent) and that given the value of c, the variables $(a_1, \ldots, a_m)$ are independent. The resulting model class is denoted by $\mathcal{M}_T$. Our assumptions can now be written as

$$P(c, a_1, \ldots, a_m \mid \mathcal{M}_T) = P(c \mid \mathcal{M}_T) \prod_{i=1}^{m} P(a_i \mid c, \mathcal{M}_T). \tag{19}$$

Suppose the special variable c has K values and each a_i has n_i values. The NML distribution for the model class $\mathcal{M}_T$ is now

$$\begin{aligned} P_{NML}(\mathbf{x}^N \mid \mathcal{M}_T) = & \left[\prod_{k=1}^{K} \left(\frac{h_k}{N} \right)^{h_k} \prod_{i=1}^{m} \prod_{k=1}^{K} \prod_{v=1}^{n_i} \left(\frac{f_{ikv}}{h_k} \right)^{f_{ikv}} \right] \\ & \cdot \frac{1}{\mathcal{R}^N_{\mathcal{M}_T,K}}, \end{aligned} \tag{20}$$

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

$$\begin{aligned} \mathcal{R}^N_{\mathcal{M}_T,K} = & \sum_{h_1+\cdots+h_K=N} \frac{N!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{N} \right)^{h_k} \\ & \cdot \prod_{i=1}^{m} \prod_{k=1}^{K} \sum_{f_{ik1}+\cdots+f_{ikn_i}=h_k} \frac{h_k!}{f_{ik1}! \cdots f_{ikn_i}!} \\ & \cdot \prod_{v=1}^{n_i} \left(\frac{f_{ikv}}{h_k} \right)^{f_{ikv}} \qquad (21) \\ = & \sum_{h_1+\cdots+h_K=N} \frac{N!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{N} \right)^{h_k} \\ & \cdot \prod_{i=1}^{m} \prod_{k=1}^{K} \mathcal{R}^{h_k}_{\mathcal{M}_{n_i}}. \qquad (22) \end{aligned}$$

It turns out [Kontkanen *et al.*, 2005] that the recursive formula (18) can be generalized also to this multi-dimensional case:

$$\begin{aligned} \mathcal{R}^N_{\mathcal{M}_T,K} = & \sum_{h_1+\cdots+h_K=N} \frac{n!}{h_1! \cdots h_K!} \prod_{k=1}^{K} \left(\frac{h_k}{N} \right)^{h_k} \\ & \cdot \prod_{i=1}^{m} \prod_{k=1}^{K} \mathcal{R}^{h_k}_{\mathcal{M}_{n_i}} \qquad (23) \\ = & \sum_{r_1+r_2=N} \frac{N!}{r_1! r_2!} \left(\frac{r_1}{N} \right)^{r_1} \left(\frac{r_2}{N} \right)^{r_2} \\ & \cdot \mathcal{R}^{r_1}_{\mathcal{M}_T,k_1} \cdot \mathcal{R}^{r_2}_{\mathcal{M}_T,k_2} \qquad (24) \\ = & \sum_{r=0}^{N} \frac{N!}{r!(N-r)!} \left(\frac{r}{N} \right)^{r} \left(\frac{N-r}{N} \right)^{N-r} \\ & \cdot \cdot \mathcal{R}^{r}_{\mathcal{M}_T,k_1} \cdot \mathcal{R}^{N-r}_{\mathcal{M}_T,k_2}, \qquad (25) \end{aligned}$$

where $k_1 + k_2 = K$.

n/k	1	2	$\cdots$	K
0	$\mathcal{R}^0_1$	$\mathcal{R}^0_2$	$\cdots$	$\mathcal{R}^0_K$
1	$\mathcal{R}^1_1$	$\mathcal{R}^1_2$	$\cdots$	$\mathcal{R}^1_K$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
N	$\mathcal{R}^N_1$	$\mathcal{R}^N_2$	$\cdots$	$\mathcal{R}^N_K$

Table 1: The regret table.

2.4 The Regret Table

As discussed in [Kontkanen *et al.*, 2005], in order to apply NML to the clustering problem, two tables of regret terms are needed. The first one consists of the one-dimensional terms $\mathcal{R}^n_{\mathcal{M}_k}$ for $n = 0, \ldots, N$ and $k = 1, \ldots, n_*$, where n_* is defined by $n_* = \max\{n_1, \ldots, n_m\}$. The second one holds the multi-dimensional regret terms needed in computing the stochastic complexity $P_{NML}(\mathbf{x}^N \mid \mathcal{M}_T)$. More precisely, this table consists of the terms $\mathcal{R}^n_{\mathcal{M}_T,k}$ for $n = 0, \ldots, N$ and $k = 1, \ldots, K$, where K is the maximum number of clusters.

The idea of the regret table can also be generalized. The natural candidate for the first dimension of the table is the size of the data. In addition to number of values or clusters, the other dimension can be, e.g., number of classes in classification tasks, or number of components in mixture modeling. The regret table is figured in Table 1.

For the single-dimensional multinomial case, the procedure of computing the regret table starts by filling the first column, i.e., the case $k = 1$. This is trivial, since clearly $\mathcal{R}^n_{\mathcal{M}_1} = 1$ for all $n = 0, \ldots, N$. To compute the column k, for $k = 2, \ldots, K$, the recursive formula (18) can be used by choosing, e.g., $k_1 = k - 1$, $k_2 = 1$. The time complexity of filling the whole table is $\mathcal{O}\left(K \cdot N^2\right)$.

The multi-dimensional case is very similar, since the recursion formula is essentially the same. The only exception is the computation of the first column. When $k = 1$, Equation (22) reduces to

$$\mathcal{R}^n_{\mathcal{M}_T,1} = \prod_{i=1}^{m} \mathcal{R}^n_{\mathcal{M}_{n_i}}, \tag{26}$$

for $n = 0, \ldots, N$. After these easy calculations, the rest of the regret table can be filled by applying the recursion (25) similarly as in the single-dimensional case. The time complexity of calculating the multi-dimensional regret table is also $\mathcal{O}\left(K \cdot N^2\right)$.

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

3 Fast Convolution

As mentioned in [Koivisto, 2004], the so-called *fast convolution* algorithm can be used to derive very efficient methods for regret computation. In this section, we will present a version suitable for computing regret tables. We will start with some mathematical background, and then proceed by deriving the fast NML algorithm.

3.1 Discrete Fourier Transform

Consider a finite-length sequence of real or complex numbers $\mathbf{a} = (a_0, a_1, \ldots, a_{N-1})$. The *Discrete Fourier Transform (DFT)* of $\mathbf{a}$ is defined as a new sequence $\mathbf{A}$ with

$$A_n = \sum_{h=0}^{N-1} a_h \cdot e^{2\pi i h n/N} \tag{27}$$

$$= \sum_{h=0}^{N-1} a_h \cdot \left(\cos \frac{2\pi h n}{N} + i \sin \frac{2\pi h n}{N}\right), \tag{28}$$

for $n = 0, \ldots, N-1$. A very intuitive explanation of the DFT is presented in [Wilf, 2002], where it is shown that if the original sequence is interpreted as the coefficients of a polynomial, the Fourier transformed sequence is obtained by evaluating the values of this polynomial at certain points on the complex unit circle. See [Wilf, 2002] for details.

To recover the original sequence $\mathbf{a}$ given the transformed sequence $\mathbf{A}$, the Discrete *inverse* Fourier Transform (DFT^{-1}) is used. The definition of DFT^{-1} is very similar to DFT, and it is given by

$$a_n = \frac{1}{N} \sum_{h=0}^{N-1} A_h \cdot e^{-2\pi i h n/N} \tag{29}$$

$$= \frac{1}{N} \sum_{h=0}^{N-1} A_h \cdot \left(\cos \frac{2\pi h n}{N} - i \sin \frac{2\pi h n}{N}\right), \tag{30}$$

for $n = 0, \ldots, N-1$.

A trivial algorithm for computing the discrete Fourier transform of length N takes time $\mathcal{O}\left(N^2\right)$. However, by means of the classic *Fast Fourier Transform (FFT)* algorithm (see, e.g., [Wilf, 2002]), this can be improved to $\mathcal{O}\left(N \log N\right)$. As we will soon see, the FFT algorithm is the basis of the fast regret table computation method.

3.2 The Convolution Theorem

A mathematical concept of *convolution* turns out to be a key element in the derivation of the fast NML algorithm. In this section, we will briefly review basic properties of convolution that are relevant to the discussion of the rest of the section.

Let $\mathbf{a} = (a_0, a_1, \ldots, a_{N-1})$ and $\mathbf{b} = (b_0, b_1, \ldots, b_{N-1})$ be two sequences of length N. The convolution of $\mathbf{a}$ and $\mathbf{b}$ is defined as a sequence $\mathbf{c}$,

$$\mathbf{c} = \mathbf{a} * \mathbf{b} \tag{31}$$

$$= (c_0, c_1, \ldots, c_{N-1}), \tag{32}$$

where

$$c_n = \sum_{h=0}^{n} a_h b_{n-h}, \; n = 0, \ldots, N-1. \tag{33}$$

Note that the convolution is mathematically equivalent to polynomial multiplication if the contents of the sequences are interpreted as the coefficients of the polynomials.

A direct computation of the convolution (33) clearly takes time $\mathcal{O}\left(N^2\right)$. The *convolution theorem* shows how to compute convolution via the discrete Fourier transform: Let $\mathbf{a} = (a_0, a_1, \ldots, a_{N-1})$ and $\mathbf{b} = (b_0, b_1, \ldots, b_{N-1})$ be two sequences. The convolution of $\mathbf{a}$ and $\mathbf{b}$ can be computed as

$$\mathbf{c} = \mathbf{a} * \mathbf{b} = \mathrm{DFT}^{-1}(\mathrm{DFT}(\mathbf{a}) \cdot \mathrm{DFT}(\mathbf{b})), \tag{34}$$

where all the vectors are zero padded to length $2N$, and the multiplication is component-wise. In other words, the Fourier transform of the convolution of two sequences is equal to the product of the transforms of the individual sequences. Since both the DFT and DFT^{-1} can be computed in time $\mathcal{O}\left(N \log N\right)$ via the FFT algorithm, and the multiplication in (34) only takes time $\mathcal{O}\left(N\right)$, it follows that the time complexity of computing the convolution sequence via DFT is $\mathcal{O}\left(N \log N\right)$.

3.3 The Fast NML Algorithm

In this section we will show how the fast convolution algorithm can be used to derive a very efficient method for the regret table computation. The new method replaces the recursion formulas (18) and (25) discussed in the previous section.

Our goal here is to calculate the column k of the regret table given the first $k-1$ columns. Let us define two sequences $\mathbf{a}$ and $\mathbf{b}$ by

$$a_n = \frac{n^n}{n!} \mathcal{R}_{k-1}^n, \;\; b_n = \frac{n^n}{n!} \mathcal{R}_1^n. \tag{35}$$

Evaluate now the convolution of $\mathbf{a}$ and $\mathbf{b}$,

$$(\mathbf{a} * \mathbf{b})_n = \sum_{h=0}^{n} \frac{h^h}{h!} \mathcal{R}_{k-1}^h \frac{(n-h)^{n-h}}{(n-h)!} \mathcal{R}_1^{n-h} \tag{36}$$

$$= \frac{n^n}{n!} \sum_{h=0}^{n} \frac{n!}{h!(n-h)!} \left(\frac{h}{n}\right)^h \left(\frac{n-h}{n}\right)^{n-h} \cdot \mathcal{R}_{k-1}^h \mathcal{R}_1^{n-h} \tag{37}$$

$$= \frac{n^n}{n!} \mathcal{R}_k^n, \tag{38}$$

where the last equality follows from the recursion formulas (18) and (25). This derivation shows that the column k can be computed by first evaluating the convolution (38), and then multiplying each term by $n!/n^n$.

It is now clear that by computing the convolutions via the DFT method discussed in the previous section, the time complexity of computing the whole regret table drops to $\mathcal{O}(N \log N \cdot K)$. This is a major improvement over $\mathcal{O}\left(N^2 \cdot K\right)$ obtained by the recursion method of Section 2.

4 Conclusion And Future Work

In this paper we discussed efficient computation algorithms for normalized maximum likelihood computation in the case of multinomial data. The focus was on the computation of the regret table needed by many applications. We showed how advanced mathematical techniques such as discrete Fourier transform and convolution can be applied to the problem.

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

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

Acknowledgements

This work was supported in part by the Academy of Finland under the projects Minos and Civi and by the National Technology Agency under the PMMA project. In addition, this work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

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

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

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

[Johnson *et al.*, 1997] N.L. Johnson, S. Kotz, and N. Balakrishnan. *Discrete Multivariate Distributions*. John Wiley & Sons, 1997.

[Koivisto, 2004] M. Koivisto. *Sum-Product Algorithms for the Analysis of Genetic Risks*. PhD thesis, Report A-2004-1, Department of Computer Science, University of Helsinki, 2004.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[Wilf, 2002] H. S. Wilf. *Algorithms and Complexity*. A K Peters, Ltd., 2002.

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