International Symposium on Information Theory and its Applications, ISITA2008
Auckland, New Zealand, 7-10, December, 2008

On Recurrence Formulas for Computing the Stochastic Complexity

Tommi Mononen and Petri Myllymäki

Helsinki Institute for Information Technology (HIIT), University of Helsinki, Finland
E-mail: <firstname>.<lastname>@hiit.fi

Abstract

Stochastic complexity is a criterion that can be used for model selection and other statistical inference tasks. Many model families, like Bayesian networks, use multinomial variables as their basic components. There now exists new efficient computation methods, based on generating functions, for computing the stochastic complexity in the multinomial case. However, the theoretical background behind these methods has not been been extensively formalized before. In this paper we define a bivariate generating function framework, which makes the problem setting more comprehensible. Utilizing this framework, we derive a new recurrence relation over the values of a multinomial variable, and show how to apply the recurrence for computing the stochastic complexity. Furthermore, we show that there cannot be a generic homogeneous linear recurrence over data size. We also suggest that the presented form of the marginal generating function, which is valid in the multinomial case, may also generalize to more complex cases.

1. Introduction

Minimum Description Length (MDL) is an information-theoretic principle for statistical inference [12]. A central concept in this framework is *stochastic complexity*. Given a parametric model $\mathcal{M}$, the stochastic complexity of a discrete observed data matrix $\mathbf{x}^n$ with n data vectors is

$$SC(\mathbf{x}^n \mid \mathcal{M}) = -\log P_{NML}(\mathbf{x}^n \mid \mathcal{M}) \qquad (1)$$

where

$$P_{NML}(\mathbf{x}^n \mid \mathcal{M}) = \frac{P(\mathbf{x}^n \mid \hat{\theta}(\mathbf{x}^n), \mathcal{M})}{\sum_{\mathbf{y}^n} P(\mathbf{y}^n \mid \hat{\theta}(\mathbf{y}^n), \mathcal{M})}. \qquad (2)$$

The above probability, where $\hat{\theta}(\mathbf{x}^n)$ denotes the maximum likelihood parameters of the model, defines the normalized maximum likelihood (NML) distribution [14]. The model selection task is to search from a fixed model family (a set of parametric models with varying complexity) the model minimizing the stochastic complexity (1).

Many probabilistic models, like for example Bayesian networks, use the multinomial variable as an important building block. However, even in this simple case computing the stochastic complexity is difficult, as the denominator of (2) (denoted in the sequel by $\mathcal{C}(L, n)$), requires summation over all the data tables $\mathbf{y}^n$ that are of the same size as our observed data. For a multinomial variable the normalizing sum is

$$\mathcal{C}(L, n) = \sum_{h_1+\cdots+h_L=n} \frac{n!}{h_1! \cdots h_L!} \prod_{k=1}^{L} \left(\frac{h_k}{n}\right)^{h_k}, \qquad (3)$$

where h_k is a number of data points assigned to the kth value and L is the number of values of the multinomial variable. Using this definition directly is obviously not feasible, but there now exist several efficient linear time algorithms for this task [6, 11] and also a very good asymptotic approximation [4] and for fixed precision there is even a sub-linear time computation method [9]. Derivation of efficient computation methods for the multinomial normalizing sum utilize the one-parameter generating function family:

$$\mathcal{B}^L(z) = \left(\frac{1}{1 - T(z)}\right)^L = \sum_{n=0}^{\infty} \mathcal{C}(L, n) n^n \frac{z^n}{n!}, \qquad (4)$$

where $T(z)$ is the tree function [5, 4]. For example, there exists an efficient recurrence formula

$$\mathcal{C}(L, n) = \mathcal{C}(L-1, n) + \left(\frac{n}{L-2}\right) \mathcal{C}(L-2, n), \qquad (5)$$

which can be used for computing multinomial normalizing sums, if the corresponding binomial normalizing sum ($L = 2$) is computed first, which can be done in $\mathcal{O}(n)$ time using the definition with a trivial parameter substitution trick [6].

However, the generating function and the recurrence formula seem to be mismatching pairs: the coefficient sequence in (4) goes over the variable n, while (5) goes over the variable L. Next we redefine the problem so that this ambiguity disappears.

978-1-4244-2069-8/08/$25.00 ©2008 IEEE

2. Bivariate Generating Function

The multinomial normalizing sum has two parameters L and n, hence we are looking for a bivariate generating function [2]. First we however define the tree function properly [5]:

$$T(z) = \sum_{n=1}^{\infty} n^{n-1} \frac{z^n}{n!} \tag{6}$$

and it satisfies

$$T(z) = ze^{T(z)}. \tag{7}$$

Now we can return to the original problem and write the bivariate generating function in the form

$$\sum_{L=0}^{\infty} \mathcal{B}^L(z) u^L = \sum_{L=0}^{\infty} \left(\frac{1}{1-T(z)} \right)^L u^L \tag{8}$$

$$= \sum_{L=0}^{\infty} \sum_{n=0}^{\infty} \mathcal{C}(L,n) n^n \frac{z^n}{n!} u^L. \tag{9}$$

We want to have (9) in closed form. Since we know that

$$\frac{1}{1-au} = 1 + au + a^2u^2 + a^3u^3 + \cdots,$$

we can just replace a by $\mathcal{B}(z)$ and get

$$f(z,u) = \sum_{L=0}^{\infty} \sum_{n=0}^{\infty} \mathcal{C}(L,n) n^n \frac{z^n}{n!} u^L \tag{10}$$

$$= \frac{1}{1 - \frac{u}{1-T(z)}} = \frac{T(z)-1}{T(z)-1+u}. \tag{11}$$

This function is the bivariate exponential generating function. Now we have the bivariate formal power series with two variable coefficients. If we arrange coefficients $\mathcal{C}(L,n)n^n$ into the form of a table, we can ask, which generating function generates coefficients of some row or column? We adopt terms and notation used in [2] and call these marginal generating functions *vertical* and *horizontal generating functions*. Now we start looking at how to compute the marginal generating functions of (11). The vertical generating function family is now our original function $\mathcal{B}^L(z)$, which we denote from now on by

$$f^{\langle L \rangle}(z) = \sum_{n=0}^{\infty} \mathcal{C}(L,n) n^n \frac{z^n}{n!} = \left(\frac{1}{1-T(z)} \right)^L. \tag{12}$$

The horizontal generating function family is a previously unknown function, although (5) applies to its sequence of coefficients. However, to derive this function, we need first the so called Lagrange inversion formula [13].

Proposition 1 *The Lagrange inversion formula is*

$$[z^n] G(\overline{H}(z)) = [z^n] G(z) H'(z) \left(\frac{H(z)}{z} \right)^{-n-1},$$

where $G(z)$ is any formal power series and $H(z)$ is any formal power series with the zero constant term and $\overline{H}(z)$ is the compositional inverse of $H(z)$.

Above we denoted the coefficient extraction by standard notation $[z^n]$. Now we are ready to present the previously missing marginal generating function family in a form of theorem.

Theorem 1 *The horizontal generating function family is of the form*

$$f_n(u) = \sum_{L=0}^{\infty} \mathcal{C}(L,n) n^n u^L$$

$$= n^n u \left(1 + \left(\frac{u}{1-u} \right) \sum_{L=0}^{n} \frac{n! n^{-L}}{(n-L)!} \cdot (1-u)^{-L} \right).$$

Proof 1 *We will utilize the Lagrange inversion, but first we have to find two functions that form a composite function:*

$$\frac{1}{1 - \frac{u}{1-T(z)}} = \frac{1}{1 - \frac{u}{1-\overline{H}(z)}} = G(\overline{H}(z), u),$$

thus the functions are

$$G(z,u) = \frac{1}{1 - \frac{u}{1-z}} = \frac{z-1}{z-1+u} \quad \textit{and}$$

$$\overline{H}(z) = T(z) \quad \textit{and} \quad H(z) = \frac{z}{e^z}.$$

Now we are ready to use the Lagrange inversion formula. We do the Lagrange inversion only with respect to variable z. The dummy variable u is only in the outer function, so we can write in this case

$$[z^n] \frac{1}{1 - \frac{u}{1-T(z)}} = [z^n] G(z,u) H'(z) \left(\frac{H(z)}{z} \right)^{-n-1}$$

$$= [z^n] \frac{z-1}{z-1+u} e^{-z} (1-z) \left(\frac{1}{e^z} \right)^{-n-1}$$

$$= [z^n] \frac{(z-1)^2 e^{nz}}{1-z-u}.$$

Next we will do series expansion for the last form. We split the form in two parts:

$$S(z,u) = \frac{(z-1)^2}{1-z-u} = \frac{z^2-2z+1}{1-z-u} \quad \textit{and}$$

$$R(z) = e^{nz}.$$

The series expansions with respect to variable z are

$$S(z,u) = (u+1) - z + \sum_{k=0}^{\infty} \frac{u^2}{(1-u)^{k+1}} z^k \quad \textit{and}$$

$$R(z) = \sum_{k=0}^{\infty} n^k \frac{z^k}{k!}.$$

The final step is just to use the discrete convolution formula between $S(z,u)$ and $R(z)$:

$$\begin{aligned}[z^n]S(z,u)R(z) &= \frac{n^n}{n!}(u+1) - \frac{n^{n-1}}{(n-1)!} \\ &\quad + \frac{u^2}{1-u} \sum_{L=0}^{n} \frac{n^{n-L}}{(n-L)!(1-u)^L} \\ &= \frac{n^n u}{n!} + \frac{u^2 n^n}{1-u} \sum_{L=0}^{n} \frac{n^{-L}}{(n-L)!}(1-u)^{-L},\end{aligned}$$

which is identical to the claim when multiplied by $n!$ (because the vertical generating function family is of exponential type). □

This result can be represented also in another form using confluent hypergeometric functions:

$$f_n(u) = n^n u + \left(\frac{n^n u^2}{1-u}\right) {}_2F_0\left(\begin{matrix}1, -n \\ --\end{matrix}\middle| -\frac{1}{n(1-u)}\right).$$

For further information on ${}_2F_0$ functions, see [11].

This one-parameter family does not look very nice at first sight, but if we enter some values, for example $n = 1, 2, 3$, we observe that the horizontal generating functions in the simplified form are

$$f_1(u) = \sum_{L=0}^{\infty} \mathcal{C}(L,1) 1^1 u^L = \frac{u}{(1-u)^2},$$

$$f_2(u) = \sum_{L=0}^{\infty} \mathcal{C}(L,2) 2^2 u^L = \frac{-2u(u-2)}{(1-u)^3} \quad \text{and}$$

$$f_3(u) = \sum_{L=0}^{\infty} \mathcal{C}(L,3) 3^3 u^L = \frac{3u(3u^2-10u+9)}{(1-u)^4}.$$

The formal power series coefficients of these functions give all the normalizing sums for a fixed amount of data. These example functions, and the horizontal generating functions in general, are so called rational generating functions. This observation enables us to use the huge mathematical toolbox designed for rational generating functions, and may be a reason why the multinomial normalizing sum has such nice properties with respect to variable L. Next we present direct implications of the redefinition.

2.1. Recurrence Relations over L

There are two known recurrence relations over variable L: the first one is (5) and the second one is actually a simple sequential convolution. The second method is based on the observation that if we have a formal power series raised to some positive integer power, we get expanded coefficients by computing many convolutions sequentially. This result applies to $f^{\langle L \rangle}(z)$.

However, our new horizontal generating function family gives the third recurrence relation. There is also a standard way to construct a recurrence relation for rational functions using Taylor series. One can say loosely that the numerator of a rational function gives initial conditions and the denominator gives the recurrence equation. Hence, in this case we get for $f_n(u)$ the recurrence equation

$$\sum_{j=0}^{n+1} \binom{n+1}{j} (-1)^j \mathcal{C}(L-j,n) = 0, \tag{13}$$

which we can write in a simpler operator form

$$(\nabla_L)^{n+1} \mathcal{C}(L,n) = 0, \tag{14}$$

where ∇_L is the backward difference operator with respect to variable L. We can write $\nabla_L = I - E_L^{-1}$, where I is the identity operator and E is the shift operator. If we apply the operator to $\mathcal{C}(L,n)$, the identity operator gives the same $\mathcal{C}(L,n)$, and the shift operator E_L^{-1} gives $\mathcal{C}(L-1,n)$. For example, if $n = 2$, we have

$$\begin{aligned}(\nabla_L)^3 \mathcal{C}(L,2) &= (I - E_L^{-1})^3 \mathcal{C}(L,2) \\ &= (I - 3E_L^{-1} + 3E_L^{-2} - E_L^{-3})\ \mathcal{C}(L,2) \\ &= \mathcal{C}(L,2) - 3\mathcal{C}(L-1,2) \\ &\quad + 3\mathcal{C}(L-2,2) - \mathcal{C}(L-3,2) \\ &= 0.\end{aligned}$$

The recurrence is valid for $f_2(u)$. In fact, also the general form of the backward difference operator is working. This means that we can skip coefficients and for example take every third one. So the general form of the operator is $\nabla_L^b = I - E_L^{-b}$. This leads to a fast algorithm for computing normalizing sums with huge values of L and a small fixed value of n. For example, for $L = n \cdot 2^m$, where $m \in \mathbb{N}$, we can compute the normalizing sums in following way: First compute the initial n values, then use the above recurrence with $b = 1$. After $2n$ values, use the recurrence with $b = 2$ and after $4n$ values use the recurrence with $b = 3$ etc. So we always make bigger and bigger leaps until we reach the desired target L. Of course we can modify the scheme so that we can reach any desired value of L using a similar approach.

This recurrence is not very useful in our framework: it starts to work after the number of values in a variable exceeds the number of data points. So it actually tells us how the value of the normalizing sum changes, if we add excess bins (values). What is more important is that the same recurrence applies also to more complex cases, like the Naive Bayes model discussed below.

2.2. Recurrence Relations over n

There are many different recurrence relations over L. For single horizontal generating functions we have the convolution recurrence and (13). For the whole horizontal family we have (5). However, there does not exist a single efficient recurrence formula over n. The existence of a simple recurrence formula over n would lead to efficient dynamic programming methods. Unfortunately we are not going to present any recurrence formulas over n; actually, in the following we prove that in this case, it is impossible to have a homogeneous linear recurrence relation for the family of vertical generating functions.

We start by definitions. We define the basic concepts first in the single variable case and later expand definitions to the multivariate case. A sequence is called *holonomic* if it satisfies a *homogeneous linear recurrence equation*

$$\begin{aligned} &p_0(i)a(i) + p_1(i)a(i+1) + \cdots \\ &+ p_d(i)a(i+r) = 0 \qquad n \geq 0, \quad r \in \mathbb{N}, \end{aligned} \tag{15}$$

where the terms $p_k(i)$ are polynomials, which are not all zero. Respectively, a formal power series is holonomic (D-finite) if and only if its coefficient sequence is holonomic (P-recursive). The following proposition defines holonomicity condition between ordinary and exponential generating functions [1].

Proposition 2 *The ordinary generating function* $\sum_{n=0}^{\infty} g_n z^n$ *is holonomic if and only if the exponential generating function* $\sum_{n=0}^{\infty} g_n \frac{z^n}{n!}$ *is holonomic.*

Now we are ready to give the proposition that is connected to our specific problem [3]:

Proposition 3 *Generating function* $\sum_{n=0}^{\infty} n^n z^n$ *is not holonomic.*

The two previous propositions together imply that also $f^{\langle 1 \rangle}(z)$ is non-holonomic. So there is no homogeneous linear recurrence relation, which gives the next coefficient given a fixed number of previous coefficients.

Until this point, we had a formal power series only with one variable. Now we need bivariate power series, because we want to prove that the bivariate generating function is non-holonomic. In general we have a formal multivariate power series

$$g(x_1, \ldots, x_2) = \sum_{i_1, \cdots, i_m} a(i_1, \ldots, i_m) x_1^{i_1} \cdots x_m^{i_m}.$$

The *section* of $g(x_1, \ldots, x_m)$ is a lower dimensional formal power series, where some variables have been assigned to certain values:

$$g^{1,\ldots,s}_{i_{s+1},\ldots,i_m}(x_1, \ldots, x_m) = \sum_{i_1,\ldots,i_s} a(i_1, \ldots, i_m) x_1^{i_1} \cdots x_s^{i_s},$$

where $(i_{s+1}, \ldots, i_m)$ are the assigned values. Now the definition of holonomicity for multivariate formal power series says that if $g(x_1, \ldots, x_m)$ is holonomic then all the sections must be holonomic [7]. This leads to the following theorem:

Theorem 2 *The bivariate generation function f(z,u) is non-holonomic.*

Proof 2 *All sections of a holonomic function must be holonomic. Now take the section*

$$\begin{aligned} f_1^n(z,u) &= \sum_{n=0}^{\infty} \mathcal{C}(1,n) n^n \frac{z^n}{n!} \\ &= \sum_{n=0}^{\infty} n^n \frac{z^n}{n!} = f^{\langle 1 \rangle}(z), \end{aligned}$$

which is non-holonomic. Therefore the bivariate generating function is also non-holonomic. □

If the bivariate generating function would have been holonomic, it would have implied that there is also a homogeneous linear recurrence over n. However now the situation is a bit more complicated. We cannot infer converse, meaning that there is no linear homogeneous recurrence, but we need a proposition that says how to end up to multivariate holonomic sequences [7]:

Proposition 4 *Let the sequence* $a(i_1, \ldots, i_m)$ *satisfy a system of recurrences, one for each* $j = 1 \ldots m$*, of the form*

$$\begin{aligned} &p_0^{(j)}(i_j) a(i_1, \ldots, i_m) + \sum_{l=1}^{r_j} p_l^{(j)}(i_1, \ldots, i_m) a(i_1, \ldots, i_{j-1}, \\ &i_j - l, i_{j+1}, \ldots, i_m) = 0, \end{aligned}$$

where $p_0^{(j)}$ *are nonzero polynomials of one variable. Then the sequence* $a(i_1, \ldots, i_m)$ *is holonomic.*

Intuitively this means that if there exists the above recurrence equations with respect to each variable, then we have a multivariate holonomic function. This leads to the following theorem:

Theorem 3 *For the family of vertical generating functions of $f(z,u)$ there is no recurrence equation of the form*

$$\sum_{l=0}^{r_2} p_l(L,n)\mathcal{C}(L,n-l)=0,$$

where r_2 is some non-negative integer and $p_0(L,n)$ is non-zero.

Proof 3 *We can use Proposition 4. First we notice that (5) can be written in form*

$$(L-2)\mathcal{C}(L,n)+(2-L)\mathcal{C}(L-1,n)\\ +(-n)\mathcal{C}(L-2,n)=0,$$

which means that the first recurrence equation exists. Now suppose that also the second recurrence exists. This implies that the sequence $\mathcal{C}(L,n)n^n$ is holonomic. However, this is a contradiction because we know that the sequence is not holonomic. So, the second recurrence must be false.

Our only concern is now that by Proposition 4 we have $p_0(n)$ instead of $p_0(L,n)$. However, if there would be such a recurrence, then by setting $L=1$ in the recurrence equation we would get a homogeneous linear recurrence for the non-holonomic $f^{\langle 1\rangle}(z)$, which is of course a contradiction. □

This result could have been proved without the bivariate holonomicity, but we wanted to bring insight to the problem setting. There still may be some other type of recurrence formulas, but finding one can be a rather difficult task as a general methodological foundation is mostly missing.

3. Recurrences in the Naive Bayes Case

The Naive Bayes classifier is a probabilistic model used widely in classification and clustering tasks. The model has m predictor variables with K_i outcomes for the ith variable, and one class variable with L outcomes. The predictor variables are assumed to be independent given the value of the class variable.

We already know that the convolution type recurrence works for the normalizing sums also in the Naive Bayes case [8]. Validity is easy to see from the fact that we can write the 'vertical' generating function family in the form

$$\left(\sum_{n=0}^{\infty}\mathcal{C}(K_1,n)\mathcal{C}(K_2,n)\cdots\mathcal{C}(K_m,n)n^n\frac{z^n}{n!}\right)^L\\ =\sum_{n=0}^{\infty}\mathcal{C}_{NB}(L,K_1,\ldots,K_m,n)n^n\frac{z^n}{n!}. \quad (16)$$

We just expand the power L sequentially and compute convolutions as we did in the multinomial case. A far more interesting fact is that our new recurrence formula seems to be valid also in the Naive Bayes case. In the following, we list some consequences of this conjecture.

It is hardly surprising that the operator equation works also over L. In this case the equation is

$$(\nabla_L)^{n+1}\mathcal{C}_{NB}(L,K_1,\ldots,K_m,n)=0. \quad (17)$$

An astonishing fact is that the same equation works also over the number of outcomes of the predictor variables. This is the first recurrence equation of this kind. The equation in this case is

$$(\nabla_{K_i})^{n+1}\mathcal{C}_{NB}(L,K_1,\ldots,K_i,\ldots,K_m,n)=0. \quad (18)$$

Using the conjecture we can construct 'horizontal' generating functions also for Naive Bayes models. We just need to compute some initial values and expect validity of recurrence equations to get the sought generating functions. We used Maple command `rectodiffeq` and solved the generating function from the result. In empirical tests we did not found any flaws. The generating functions seem to be the correct ones.

We give an example. Let us take a Naive Bayes model whose root node has 5 values and the three leaf nodes 3, 3 and K_3 values. For values $n=1,2,3$, the generating functions are

$$f_1(u)=45\frac{u}{(1-u)^2},$$
$$f_2(u)=\frac{405}{2}\frac{u(7u+10)}{(1-u)^3} \quad \text{and}$$
$$f_3(u)=\frac{5}{27}\frac{u(126525u^2+1152890u+497673)}{(1-u)^4}.$$

We can see that the denominators are of course same as in the multinomial case for these few instances. Moreover, also the horizontal generating functions are quite similar, only the coefficients of the numerators are different.

4. Using Horizontal Generating Functions

Let us suppose that we have precomputed the expanded form of a horizontal generating function for some model and fixed n. Now we can compute the normalizing values using convolution. The series expansion of the denominators of horizontal generating functions is

$$\frac{1}{(1-u)^{n+1}}=\sum_{j=0}^{\infty}\binom{j+n}{j}u^j=\sum_{j=0}^{\infty}h_ju^j. \quad (19)$$

and the coefficients of the series can be computed efficiently using the ratio

$$\frac{h_j}{h_{j-1}} = \frac{j+n}{j}. \tag{20}$$

We just have to do the discrete convolution for coefficient sequences of (19) and the numerator. This way we can test different number of values in a single leaf (predictor) variable of a Naive Bayes model very fast. We have also noticed that leaf nodes of Bayesian trees have similar kind of horizontal generating functions. This may generalize also to inner and root nodes, but our present tree computation algorithm does not allow us to test this hypothesis.

However, as the computation has been previously shown to involve higher level convolutions with respect to sequences of length n [8, 10], we expect there to be hidden costs. We are still hoping that there could be efficient shortcuts with respect to a tree structure, which would allow us to construct the horizontal generating functions efficiently. However, it is most likely that bivariate generating functions are not descriptive enough for more complex models, but in this case we need multivariate generating functions.

5. Conclusions

We redefined the generating function for the multinomial normalizing sum of NML. The previously unknown marginal generating function family has interesting properties and it may eventually lead to development of efficient algorithms for computing the stochastic complexity of Bayesian trees and more general Bayesian models. We also proved that for computing the multinomial normalizing sum, there does not exist a generic homogeneous linear recurrence formula over the data size.

Acknowledgments

Authors thanks Alessandro Di Bucchianico and Petri Kontkanen for fruitful discussions. This work was supported in part by the Academy of Finland under the project Civi and by the Finnish Funding Agency for Technology and Innovation under the projects Kukot and PMMA. In addition, this work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence.

References

[1] C. Banderier, M. Bousquet-Mélou, A. Denise, P. Flajolet, D. Gardy, and D. Gouyou-Beauchamps. Generating functions for generating trees. *Discrete Mathematics*, 246(1-3):29–55, March 2002.

[2] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Unpublished, 2005.

[3] S. Gerhold. *Combinatorial Sequences: Non-Holonomicity and Inequalities*. PhD thesis, Johannes Kepler University, Linz, 2005.

[4] P. Jacquet and W. Szpankowski. Markov types and minimax redundancy for Markov sources. *IEEE Transactions on Information Theory*, 50(7):1393–1402, 2004.

[5] D.E. Knuth. Convolution polynomials. *Mathematica Journal*, 2(4):67–78, Fall 1992.

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

[7] L. Lipshitz. D-finite power series. *Journal of Algebra*, 122:353–373, 1989.

[8] T. Mononen and P. Myllymäki. Fast NML computation for Naive Bayes models. In V. Corruble, M. Takeda, and E. Suzuki, editors, *Proceedings of the 10th International Conference on Discovery Science*, October 2007.

[9] T. Mononen and P. Myllymäki. Computing the multinomial stochastic complexity in sub-linear time. In *Proceedings of the 4th European Workshop on Probabilistic Graphical Models*, 2008.

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

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

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

[13] S. Roman. *The Umbral Calculus*. Dover, 2005.

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

Errata: On Recurrence Formulas for Computing the Stochastic Complexity

December 19, 2008

In Section 2.2

Equation (15): $p_0(i)a(i) + p_1(i)a(i+1) + \cdots + p_r(i)a(i+r) = 0$, $i, r \in \mathbb{N}$,

Page 4, 2nd col, row 3: $g(x_1, \ldots, x_m) = \sum_{i_1, \cdots, i_m} a(i_1, \ldots, i_m) x_1^{i_1} \cdots x_m^{i_m}$

Page 4, 2nd col, row 7: $g^{1,\ldots,s}_{i_{s+1},\ldots,i_m}(x_1, \ldots, x_s) = \sum_{i_1,\ldots,i_s} a(i_1, \ldots, i_m) x_1^{i_1} \cdots x_s^{i_s}$

Proof of Theorem 2: $f_1^1(z) = \sum_{n=0}^{\infty} \mathcal{C}(1,n) n^n \frac{z^n}{n!} = \cdots \qquad (x_1 = z,\ x_2 = u)$

Theorem 3: $\sum_{l=0}^{r_2} p_l(L,n)\ \mathcal{C}(L, n-l) \frac{(n-l)^{n-l}}{(n-l)!} = 0$

Proof of Theorem 3: ... sequence $\mathcal{C}(L,n)\frac{n^n}{n!}$ is ...

Correction of an consequence of Theorem 3: For the vertical family there cannot be a homogeneus linear recurrence equation, but this in fact does **not** prove that the same applies also for the sequence of $\mathcal{C}(L,n)$ over the variable n (The wrong consequence is mentioned in Abstract and Conclusions).