Context Weighting : General Finite Context Sources

Frans M.J. Willems, Yuri M. Shtarkov and Tjalling J. Tjalkens

Eindhoven University of Technology and Institute for Problems of Information Transmission, Moscow

Abstract

Context weighting procedures are presented for sources with models in four different classes. Although the procedures are designed for universal data compression, their generality allows application in the area of classification.

1 Introduction

Recently [7] *context tree weighting* was introduced as a sequential universal source coding method for the class of binary FSMX sources, as defined by Rissanen [5]. The idea behind weighting procedures can be summarized as follows.

The well known Elias algorithm (described in e.g. Jelinek [1]) produces for any coding distribution $P_c(x_1 \cdots x_T)$ over all binary sequences of length T, a binary prefix code with codeword lengths $L(x_1 \cdots x_T)$ that satisfy

$$L(x_1 \cdots x_T) \leq \log \frac{1}{P_c(x_1 \cdots x_T)} + 2 \text{ for all } x_1 \cdots x_T. \tag{1}$$

(We assume that the base of the $\log(\cdot)$ is 2. Codeword lengths and information quantities are expressed in bits.) If the marginals $P_c(x_1 \cdots x_t) = \sum_{x_{t+1} \cdots x_T} P_c(x_1 \cdots x_T), t = 1, \cdots, T$ are sequentially available the *arithmetic* code can be implemented sequentially. Accepting a *coding redundancy* of at most 2 bits, we are now left with the problem of finding good coding distributions P_c.

For memoryless sources with unknown parameter θ (the probability of generating a 1), it is reasonable to assign the block probability $P_c(x_1 \cdots x_T) = P_e(a, b)$ to a sequence $x_1 \cdots x_T$ containing a zeros and b ones where

$$P_e(a,b) \triangleq \frac{\frac{1}{2} \cdot \frac{3}{2} \cdot \ldots \cdot (a - \frac{1}{2}) \cdot \frac{1}{2} \cdot \frac{3}{2} \cdot \ldots \cdot (b - \frac{1}{2})}{1 \cdot 2 \cdot \ldots \cdot (a+b)} \text{ for } a > 0 \text{ and } b > 0, \text{ etc.} \tag{2}$$

This distribution, which allows sequential updating and therefore sequential coding, was suggested by Krichevsky and Trofimov [2]. It guarantees uniform convergence of the *parameter redundancy*, i.e. for any sequence $x_1 \cdots x_T$ with actual probability $P_a(x_1 \cdots x_T) = (1-\theta)^a \theta^b$, it can be shown (see [8]) that

$$\log \frac{P_a(x_1 \cdots x_T)}{P_c(x_1 \cdots x_T)} \leq \frac{1}{2} \log T + 1 \text{ for all } \theta \in [0, 1]. \tag{3}$$

In a more general setting the source is not memoryless. The distribution that the source uses to generate the next symbol $X_t, t = 1, \cdots, T$, is determined by the binary sequence $u_t(1) \cdots u_t(D)$, called the *context* of x_t. One can think of sources for which the context consists of the D most recent source outputs, thus $u_t(d) = x_{t-d}, d = 1, \cdots, D$. More general context definitions are possible, it is assumed however that the context $u_t(1) \cdots u_t(D)$ is available to the encoder at encoding time and to the decoder at decoding time of x_t.

The mapping M from the context space $\{0,1\}^D$ into the parameter-index set $\mathcal{K}$, is what we call the *model* of the source. To each parameter-index $k \in \mathcal{K}$ there corresponds a parameter $\theta(k) \in [0,1]$. The source generates X_t, with a probability of a 1 equal to $\theta(M(u_t(1) \cdots u_t(D)))$.

If we know the actual model M_a, we can partition the sequence $x_1 \cdots x_T$ in memoryless subsequences and use $P_c(x_1 \cdots x_T | M_a) = \Pi_{k \in \mathcal{K}_a} P_e(a_k, b_k)$ as a coding distribution, where a_k, resp. b_k is the number of instants t for which $x_t = 0$, resp. 1 and $M_a(u_t(1) \cdots u_t(D)) = k$. The image of $\{0,1\}^D$ under M_a is $\mathcal{K}_a$. Again this coding distribution allows sequential updating. For any sequence $x_1 \cdots x_T$, using (3) and the convexity of the $\log(\cdot)$, the parameter redundancy can now be upper bounded as

$$\log \frac{P_a(x_1 \cdots x_T)}{P_c(x_1 \cdots x_T | M_a)} \leq \frac{|\mathcal{K}_a|}{2} \log \frac{T}{|\mathcal{K}_a|} + |\mathcal{K}_a| \text{ for all } M_a \in \mathcal{M} \text{ and } \theta(k) \in [0,1], k \in \mathcal{K}_a, \tag{4}$$

where $P_a(x_1 \cdots x_T) = \Pi_{k \in \mathcal{K}_a} (1 - \theta(k))^{a_k} \theta^{b_k}(k)$ is the actual probability of $x_1 \cdots x_T$.

If the model is unknown we *weight* the coding distributions corresponding to all models M in the *model class* $\mathcal{M}$ and obtain the coding distribution $P_c(x_1 \cdots x_T) = \sum_{M \in \mathcal{M}} P(M) P_c(x_1 \cdots x_T | M)$. Here $P(M)$ is the a priori probability that is assigned to the model M in class $\mathcal{M}$. For any sequence $x_1 \cdots x_T$ the *model redundancy* can now be upper bounded as

$$\log \frac{P_c(x_1 \cdots x_T | M_a)}{P_c(x_1 \cdots x_T)} \leq \log \frac{1}{P(M_a)} \text{ for all } M_a \in \mathcal{M}. \tag{5}$$

The *total cumulative redundancy* is equal to the sum of the (cumulative) model, parameter and coding redundancies. Using (1),(4), and (5) we can upper bound this total redundancy for any sequence $x_1 \cdots x_T$ in the following way :

$$\begin{aligned} & L(x_1 \cdots x_T) - \log \frac{1}{P_a(x_1 \cdots x_T)} = \\ & \quad \log \frac{P_c(x_1 \cdots x_T | M_a)}{P_c(x_1 \cdots x_T)} + \log \frac{P_a(x_1 \cdots x_T)}{P_c(x_1 \cdots x_T | M_a)} + L(x_1 \cdots x_T) - \log \frac{1}{P_c(x_1 \cdots x_T)} \\ & \quad \leq \log \frac{1}{P(M_a)} + \frac{|\mathcal{K}_a|}{2} \log \frac{T}{|\mathcal{K}_a|} + |\mathcal{K}_a| + 2. \end{aligned} \tag{6}$$

This holds for all models $M_a \in \mathcal{M}$ and parameters $\theta(k) \in [0,1], k \in \mathcal{K}_a$. Rewriting this bound, and taking the minimum over all actual source models and parameters we obtain

$$\begin{aligned} L(x_1 \cdots x_T) \leq \min_{M_a \in \mathcal{M}, \theta(k) \in [0,1], k \in \mathcal{K}_a} \{ & \log \frac{1}{P_a(x_1 \cdots x_T)} \quad + \\ & \log \frac{1}{P(M_a)} + \frac{|\mathcal{K}_a|}{2} \log \frac{T}{|\mathcal{K}_a|} \quad + \quad |\mathcal{K}_a| + 2 \}. \end{aligned} \tag{7}$$

Assuming that these upper bounds are (more or less) tight we can conclude that context weighting methods minimize the *total description length* of a sequence.

In the next sections we consider four model classes. We show that for each of these classes there exist natural a priori distributions over the models, that allow efficient (sequential) computation of the corresponding weighted probability $P_c(x_1 \cdots x_T)$.

2 Splittings

It is natural to view a model as a partition of the set of all contexts $\{0,1\}^D$ into $|\mathcal{K}|$ cells, one for each parameter $\theta(k), k \in \mathcal{K}$. Since each partition can be generated by a sequence of *splittings*, these are partitions into two cells, a model partitions subsets of $\{0,1\}^D$ into smaller subsets, performing binary splittings only. The model class determines which splittings are allowed, and therefore what the structure of the resulting context sets is. A splitting which is always possible is the *void* splitting, corresponding to the assumption that all contexts in the considered subset are mapped into the same parameter. Further splitting is unnecessary then. Assuming that all possible splittings are equally likely, we can define a code that specifies a model. This code is defined recursively, starting from the complete context set $\{0,1\}^D$. For each context subset the code is the concatenation of the code that specifies the splitting, followed by the two codes for the (complementary) subsets that have resulted from the splitting, only when the splitting was non-void however.

Example : Consider the case where $D = 3$. We assume for the model that $M(000) = M(001) = M(010) = \alpha, M(011) = M(100) = \beta, M(101) = M(110) = \alpha$, and $M(111) = \beta$. If we allow arbitrary splitting (this corresponds to Model Class I as we will see soon) there are 127 possible splittings of 8 contexts plus the void splitting. Therefore we need $\log 128 = 7$ bits to specify this first splitting. After this splitting there are two context sets $\{000, 001, 010, 101, 110\}$ and $\{011, 100, 111\}$ we have to deal with. The code for each of these subsets is the code for the void splitting however. We need 4 bits for the first subset and 2 bits for the second one. In total 13 bits are needed to describe the model M if arbitrary splitting is allowed, resulting in $|\mathcal{K}| = 2$ parameter-indices.

In the next section we will see that these splittings lead to efficient weighting methods.

3 Model Classes

Weighting is assigning probabilities to subsequences corresponding to context subsets. The subsequence corresponding to a subset $\mathcal{S}$ of the set of all contexts $\{0,1\}^D$ is the concatenation of all source symbols with contexts in $\mathcal{S}$. The problem now is whether this subsequence should be considered memoryless or whether the context set $\mathcal{S}$ (and also the subsequence) should be further splitted. For a memoryless subsequence we can use the estimator $P_e(\mathcal{S}) \triangleq P_e(a_{\mathcal{S}}, b_{\mathcal{S}})$ where $a_{\mathcal{S}}$, resp. $b_{\mathcal{S}}$ is the number of instants t for which $u_t(1) \cdots u_t(D) \in \mathcal{S}$ and $x_t = 0$, resp. 1. If a certain splitting is necessary we should multiply the estimated (weighted) probabilities for the subsequences corresponding to the complementary subsets that result from the splitting, $P_w(\mathcal{T})$ and $P_w(\mathcal{S} - \mathcal{T})$, with

each other. Since none of the alternatives is more favorite than the others we just average the estimated probabilities corresponding to all the splittings, including the void splitting.

3.1 Class I : Arbitrary Splitting

The number of splittings of $\mathcal{S}$ including the void splitting is in this case equal to $2^{|\mathcal{S}|-1}$. The recursive weighting algorithm for this most general form of splitting is defined by

$$P_w(\mathcal{S}) \triangleq \frac{P_e(\mathcal{S}) + \sum_{\mathcal{T}\subset\mathcal{S}, 0^D\in\mathcal{T}} P_w(\mathcal{T})P_w(\mathcal{S}-\mathcal{T})}{2^{|\mathcal{S}|-1}}, \tag{8}$$

where it is understood that $\mathcal{T} \neq \mathcal{S}$. The weighted probability $P_w(\{0,1\}^D)$ can be used as coding probability. What we mean by this is the following. Suppose data (context and source symbols) are processed in a structure of records, one for each subset of $\{0,1\}^D$, for $1, \cdots, t-1$. Then $P_c(x_1 \cdots x_{t-1}) = P_w(\{0,1\}^D)$. Now the context $u_t(1) \cdots u_t(D)$ becomes available to encoder and decoder. Updating the structure of records with $x_t = 0$ would yield the block probability $P_c(x_1 \cdots x_{t-1}0)$ and updating with $x_t = 1$ gives $P_c(x_1 \cdots x_{t-1}1)$. These probabilities can be used for sequential encoding and decoding. It is easily checked that $P_c(x_1 \cdots x_{t-1}0) + P_c(x_1 \cdots x_{t-1}1) = P_c(x_1 \cdots x_{t-1})$.

Inspection shows that the models are weighted with an a priori distribution which is equal to the sum of the probabilities induced by the lengths of all codes that specify the model, i.e. the partition of $\{0,1\}^D$.

Example : The model redundancy of our model is at most 13 bits. There is only one sequence of splittings that specifies the model. Therefore

$$\begin{aligned} P_w(\{0,1\}^3) &\geq 2^{-7} P_w(\{000,001,010,101,110\}) P_w(\{011,100,111\}) \\ &\geq 2^{-7}2^{-4} P_e(\{000,001,010,101,110\}) 2^{-2} P_e(\{011,100,111\}) \\ &= 2^{-13} P_e(\{000,001,010,101,110\}) P_e(\{011,100,111\}). \end{aligned} \tag{9}$$

The number of parameters-indices of our model in class I is $|\mathcal{K}| = 2$. In this class our model has the lowest possible parameter redundancy.

3.2 Class II : Lexicographical Splitting

Define $B(u_1 \cdots u_D) \triangleq \sum_{d=1,D} u_d 2^{D-d}$ as the *index* of the context $u_1 \cdots u_D$. This index determines a lexicographical ordering over the set of contexts, but any other ordering would do as well. For $0 \leq i < j \leq 2^D$ define $\mathcal{S}_{i,j}$ as the set of all contexts with an index between i and j i.e. $\mathcal{S}_{i,j} \triangleq \{s \in \{0,1\}^D | i \leq B(s) < j\}$.

The recursive weighting procedure for lexicographical splitting is defined by

$$P_w(\mathcal{S}_{i,j}) \triangleq \frac{P_e(\mathcal{S}_{i,j}) + \sum_{k=i+1,j-1} P_w(\mathcal{S}_{i,k}) P_w(\mathcal{S}_{k,j})}{j-i}. \tag{10}$$

Probability $P_w(S_{0,2^D}) = P_w(\{0,1\}^D)$ can be used for sequential encoding and decoding.

Example : The model redundancy is at now most 8.6 bits. This follows from the decomposition :

$$
\begin{aligned}
P_w(\mathcal{S}_{0,8}) &\geq \frac{1}{8}(P_w(\mathcal{S}_{0,3})P_w(\mathcal{S}_{3,8}) + P_w(\mathcal{S}_{0,5})P_w(\mathcal{S}_{5,8}) + P_w(\mathcal{S}_{0,7})P_w(\mathcal{S}_{7,8})) \\
&\geq \frac{1}{8}(\frac{1}{3}P_e(\mathcal{S}_{0,3})\frac{1}{5}(P_w(\mathcal{S}_{3,5})P_w(\mathcal{S}_{5,8}) + P_w(\mathcal{S}_{3,7})P_w(\mathcal{S}_{7,8})) \\
&\quad +\frac{1}{5}P_w(\mathcal{S}_{0,3})P_w(\mathcal{S}_{3,5})\frac{1}{3}P_w(\mathcal{S}_{5,7})P_w(\mathcal{S}_{7,8}) \\
&\quad +\frac{1}{7}(P_w(\mathcal{S}_{0,3})P_w(\mathcal{S}_{3,7}) + P_w(\mathcal{S}_{0,5})P_w(\mathcal{S}_{5,7}))\frac{1}{1}P_e(\mathcal{S}_{7,8})) \\
&\geq \cdots \geq \frac{13}{5040}P_e(\mathcal{S}_{0,3})P_e(\mathcal{S}_{3,5})P_e(\mathcal{S}_{5,7})P_e(\mathcal{S}_{7,8}). \qquad (11)
\end{aligned}
$$

There are five sequences of splittings that specify our model in class II. One of these is $\mathcal{S}_{0,8} \rightsquigarrow \mathcal{S}_{0,5}\mathcal{S}_{5,8} \rightsquigarrow \mathcal{S}_{0,3}\mathcal{S}_{3,5}\mathcal{S}_{5,7}\mathcal{S}_{7,8}$. The length of the specification code for this splitting is $\log 8 + (\log 5 + (\log 3 + \log 2)) + (\log 3 + (\log 2 + \log 1)) = \log 1440 = 10.5$ bits. The induced probability $1/1440$.

The number of parameters indices of our model in this class is $|\mathcal{K}| = 4$. This leads to a higher parameter redundancy than for arbitrary splittings.

3.3 Class III : Arbitrary Position Splitting

In this class (and also in class IV) context subsets are splitted according to the value of a context digit. The position of this context digit can be arbitrary, i.e. in $\{1, \cdots, D\}$ in class III. A subset is determined by the set $\mathcal{P}$ of positions and the sequence of values $\prod_{i\in\mathcal{P}} v_i$ at these positions, hence $\mathcal{S}_{\mathcal{P},\prod_{i\in\mathcal{P}} v_i} \stackrel{\Delta}{=} \{u_1 \cdots u_D | u_i = v_i, i \in \mathcal{P}\}$.

Recursive weighting for arbitrary position splitting is defined by

$$
\begin{aligned}
&P_w(\mathcal{S}_{\mathcal{P},\prod_{i\in\mathcal{P}} v_i}) \stackrel{\Delta}{=} \\
&\quad \frac{P_e(\mathcal{S}_{\mathcal{P},\prod_{i\in\mathcal{P}} v_i}) + \sum_{p\in\{1,\cdots,D\},p\notin\mathcal{P}} P_w(\mathcal{S}_{\mathcal{P}\cup\{p\},(\prod_{i\in\mathcal{P}} v_i)\times 0})P_w(\mathcal{S}_{\mathcal{P}\cup\{p\},(\prod_{i\in\mathcal{P}} v_i)\times 1})}{D - |\mathcal{P}| + 1}, \qquad (12)
\end{aligned}
$$

for position sets $\mathcal{P} \neq \{1, \cdots, D\}$. For subsets containing a single context, i.e. subsets for which all positions are specified, we have $P_w(\mathcal{S}_{\{1,\cdots,D\},v_1\cdots v_D}) = P_e(\mathcal{S}_{\{1,\cdots,D\},v_1\cdots v_D}) = P_e(\{v_1 \cdots v_D\})$. The weighted probability $P_w(S_{\phi,\lambda}) = P_w(\{0,1\}^D\})$ can be used for sequential encoding and decoding. Here ϕ is the empty set, and λ the empty sequence.

Example : In this class the model redundancy is 8.2 bits :

$$
\begin{aligned}
&P_w(\mathcal{S}_{\phi,\lambda}) \geq \frac{1}{4}(P_w(\mathcal{S}_{\{2\},0})P_w(\mathcal{S}_{\{2\},1}) + P_w(\mathcal{S}_{\{3\},0})P_w(\mathcal{S}_{\{3\},1})) \\
&\geq \frac{1}{4}(\frac{1}{3}(P_w(\mathcal{S}_{\{2,3\},00})P_w(\mathcal{S}_{\{2,3\},01}))\frac{1}{3}(P_w(\mathcal{S}_{\{2,3\},10})P_w(\mathcal{S}_{\{2,3\},11})) \\
&\quad +\frac{1}{3}(P_w(\mathcal{S}_{\{3,2\},00})P_w(\mathcal{S}_{\{3,2\},01}))\frac{1}{3}(P_w(\mathcal{S}_{\{3,2\},10})P_w(\mathcal{S}_{\{3,2\},11}))) \\
&= \frac{2}{4}(\frac{1}{3}(P_w(\mathcal{S}_{\{2,3\},00})P_w(\mathcal{S}_{\{2,3\},01}))\frac{1}{3}(P_w(\mathcal{S}_{\{2,3\},10})P_w(\mathcal{S}_{\{2,3\},11}))) \\
&\geq \frac{2}{4}(\frac{1}{3}(\frac{1}{2}(P_w(\mathcal{S}_{\{2,3,1\},000})P_w(\mathcal{S}_{\{2,3,1\},001}))\frac{1}{2}P_e(\mathcal{S}_{\{2,3\},01}))\frac{1}{3}(\frac{1}{2}P_e(\mathcal{S}_{\{2,3\},10})\frac{1}{2}P_e(\mathcal{S}_{\{2,3\},11})))
\end{aligned}
$$

$$\geq \frac{2}{4}(\frac{1}{3}(\frac{1}{2}(P_e(\mathcal{S}_{\{2,3,1\},000})P_e(\mathcal{S}_{\{2,3,1\},001}))\frac{1}{2}P_e(\mathcal{S}_{\{2,3\},01}))\frac{1}{3}(\frac{1}{2}P_e(\mathcal{S}_{\{2,3\},10})\frac{1}{2}P_e(\mathcal{S}_{\{2,3\},11})))$$
$$= \frac{1}{288}P_e(\{000\})P_e(\{100\})P_e(\{001,101\})P_e(\{010,110\})P_e(\{011,111\}). \quad (13)$$

There are two sequences of splittings that specify our model in class III. One starts with a splitting at position 2 and then proceeds with a splitting at position 3, the other starts with a splitting at 3 and then a next one at 2. Both sequences of splittings induce a probability 1/576, so the total a priori probability of this model in class III is 1/288. The number of indices in $\mathcal{K}$ is 5, which is (again) higher as the number for the previously described class.

3.4 Class IV : Next Position Splitting

In class IV context subsets are splitted according to the value of the "next" context digit. Subsets are determined by the sequence of values $\prod_{i=1,d} v_i$ at "previously" splitted positions, thus $\mathcal{S}_{\prod_{i=1,d} v_i} \triangleq \{u_1 \cdots u_D | u_i = v_i, i = 1, \cdots, d\}$.

The recursive weighting procedure for next position splitting is given by

$$P_w(\mathcal{S}_{\prod_{i=1,d} v_i}) \triangleq \frac{P_e(\mathcal{S}_{\prod_{i=1,d} v_i}) + P_w(\mathcal{S}_{(\prod_{i=1,d} v_i)\times 0})P_w(\mathcal{S}_{(\prod_{i=1,d} v_i)\times 1})}{2}, \quad (14)$$

for $d = 0, 1, \cdots, D-1$. For subsets containing a single context only, i.e. subsets for $d = D$, we have $P_w(\mathcal{S}_{v_1 \cdots v_D}) = P_e(\mathcal{S}_{v_1 \cdots v_D}) = P_e(\{v_1 \cdots v_D\})$. The weighted probability $P_w(S_\lambda) = P_w(\{0,1\}^D)$ can be used for sequential encoding and decoding.

Example : Our model costs 7 bits in class IV. The number of indices in $\mathcal{K}$ is 7. We decompose as follows :

$$P_w(\mathcal{S}_\lambda) \geq \frac{1}{2}(P_w(\mathcal{S}_0)P_w(\mathcal{S}_1)) \geq \frac{1}{2}(\frac{1}{2}(P_w(\mathcal{S}_{00})P_w(\mathcal{S}_{01}))\frac{1}{2}(P_w(\mathcal{S}_{10})P_w(\mathcal{S}_{11})))$$
$$\geq \frac{1}{2}(\frac{1}{2}(\frac{1}{2}P_e(\mathcal{S}_{00})\frac{1}{2}(P_w(\mathcal{S}_{010})P_w(\mathcal{S}_{011})))\frac{1}{2}(\frac{1}{2}(P_w(\mathcal{S}_{100})P_w(\mathcal{S}_{101}))\frac{1}{2}(P_w(\mathcal{S}_{110})P_w(\mathcal{S}_{111}))))$$
$$= \frac{1}{2}(\frac{1}{2}(\frac{1}{2}P_e(\mathcal{S}_{00})\frac{1}{2}(P_e(\mathcal{S}_{010})P_e(\mathcal{S}_{011})))\frac{1}{2}(\frac{1}{2}(P_e(\mathcal{S}_{100})P_e(\mathcal{S}_{101}))\frac{1}{2}(P_e(\mathcal{S}_{110})P_e(\mathcal{S}_{111}))))$$
$$= \frac{1}{128}P_e(\{000,001\})P_e(\{010\})P_e(\{011\})P_e(\{100\})P_e(\{101\})P_e(\{110\})P_e(\{111\}). \quad (15)$$

In class IV there is always only one sequence of splittings that specifies a model.

4 Simulations

In our example we considered a certain model. We have simulated a source that generates information according to this model for a context definition $u_t(d) = x_{t-d}, d = 1, 2, 3$. The parameters were chosen $\theta_\alpha = 0.8$ and $\theta_\beta = 0.1$.

The source produced a sequence of $T = 2^{16}$ binary digits (after having generated 3 digits that were necessary to form the first three contexts). We computed for this

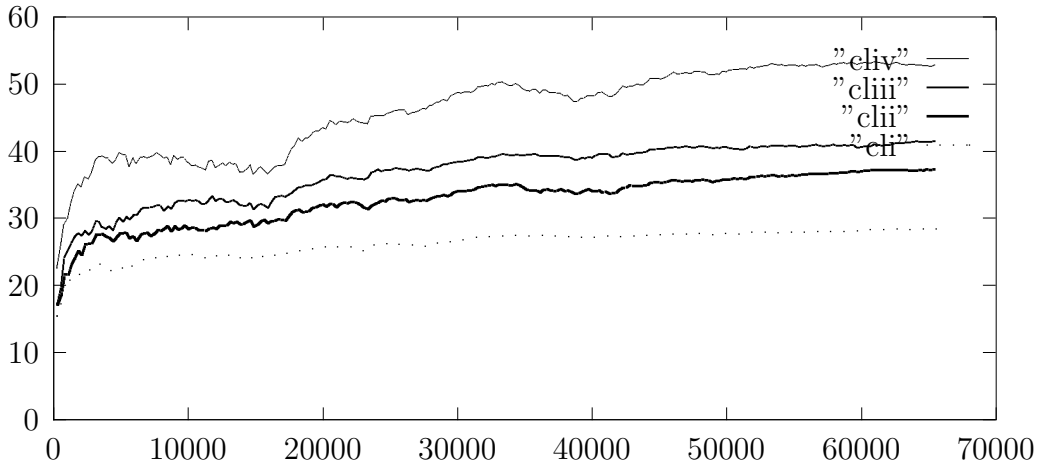

Figure 1: Cumulative redundancies in bits for $t = 1, 2, \cdots, 2^{16}$.

sequence for each of the four procedures defined by (8), (10), (12), and (14), for $t = 1, 2, \cdots, T$ the cumulative redundancy $\log P_a(x_1 \cdots x_t)/P_c(x_1 \cdots x_t)$, which is the total redundancy under the assumption that there is no coding redundancy. The results are plotted in the figure.

In the previous section we have seen that the model redundancies for our model in each of the four classes are upper bound by 13.0, 8.6, 8.2 reps. 7.0 bits. Upper bound (4) leads to parameter redundancies that can not exceed 17.0, 32.0, 39.2, reps. 53.2 bits for our model in the four different classes. The total redundancies are therefore upper bounded by 30.0, 40.6, 47.4, resp. 60.2 bits. The figure shows that the computed redundancies are close to these bounds.

5 Remarks

In the previous section we have described four model classes together with their weighting algorithms. All these methods achieve Rissanen's asymptotic lower bound on the redundancy [4]. It can also be shown that when some other code gives lower redundancies than our code for certain sources, it must yield higher redundancies for other sources in the class (see [8]).

Although we have only considered binary sources and binary contexts here, it is straightforward to generalize to non-binary cases. In our presentation of the weighting algorithms we assume infinite precision arithmetic. Modifications exist however, that can be implemented on fixed register length machines.

As a final remark we mention the application of weighting to classification based on Rissanen's *minimum description length principle* (see [6] and also Quinlan and Rivest [3]). Considering the attributes, or tests, of an object t as its context $u_t(1) \cdots u_t(D)$ and the class of the object as source output x_t, classification can be regarded as a source

coding problem.

The flexibility of weighting, allows us to describe efficient methods for producing minimum description length classification trees. Observe that when we take the maximum weighted probability over the splittings of a context subset, instead of adding them together, and divide by the total number of splittings, we obtain the minimum description length of the data as in (7). Tracking this procedure yields the minimum description length model. Combinations of our weighting methods for the different classes, lead to interesting classification procedures, even for attributes that take values in "large" alphabets. Note that the algorithm for class III selects the positions (attributes) which gives the highest reduction of the description length, while class II methods can be used to find the most effective thresholds in large attribute alphabets. The fact that there exist elegant weighting methods to treat missing attributes, demonstrates once more the flexibility of weighting.

References

[1] F. Jelinek, *Probabilistic Information Theory,* New York: McGraw-Hill, 1968, pp. 476-489.

[2] R.E. Krichevsky and V.K. Trofimov, "The Performance of Universal Encoding," *IEEE Trans. Inform. Theory,* vol. IT-27, pp. 199-207, March 1981.

[3] J.R. Quinlan and R.L. Rivest, "Inferring Decision Trees Using the Minimum Description Length Principle," *Inform. and Comput.,* vol. 80, pp. 227-248, 1989.

[4] J. Rissanen, "Universal Coding, Information, Prediction, and Estimation," *IEEE Trans. Inform. Theory,* vol. IT-30, pp. 629-636, July 1984.

[5] J. Rissanen, "Complexity of Strings in the Class of Markov Sources," *IEEE Trans. Inform. Theory,* vol. IT-32, pp. 526-532, July 1986.

[6] J. Rissanen, *Stochastic Complexity in Statistical Inquiry.* Singapore : World Scientific Publ. Co., 1989.

[7] F.M.J. Willems, Y.M. Shtarkov and Tj.J. Tjalkens, "Context Tree Weighting : A Sequential Universal Source Coding Procedure for FSMX Sources," IEEE Int. Symp. on Inform. Theory, San Antonio, Texas, Jan. 17-22, 1993, p. 59.

[8] F.M.J. Willems, Y.M. Shtarkov and Tj.J. Tjalkens, "Context Tree Weighting : Redundancy Bounds and Optimality," submitted for presentation at the *6th Swedish-Russian Workshop on Information Theory*, Mölle, Sweden, Aug. 22-27, 1993.