

# Context-Tree Weighting and Maximizing: Processing Betas

Frans M.J. Willems  
Eindhoven University of Technology  
Eindhoven, The Netherlands  
Email: f.m.j.willems@tue.nl

Tjalling J. Tjalkens  
Eindhoven University of Technology  
Eindhoven, The Netherlands  
Email: t.j.tjalkens@tue.nl

Tanya Ignatenko  
Eindhoven University of Technology  
Eindhoven, The Netherlands  
Email: t.ignatenko@tue.nl

**Abstract**—The context-tree weighting method (Willems, Shtarkov, and Tjalkens [1995]) is a sequential universal source coding method that achieves the Rissanen lower bound [1984] for tree sources. The same authors also proposed context-tree maximizing, a two-pass version of the context-tree weighting method [1993]. Later Willems and Tjalkens [1998] described a method based on ratios (betas) of sequence probabilities that can be used to reduce the storage complexity of the context-tree weighting method. These betas can be applied to express a posteriori model probabilities in a recursive way (Willems, Nowbahkt-irani, Volf [2001]). In the present paper we present new results related to betas. These results provide a new view on the relation between context-tree weighting and maximizing.

## I. INTRODUCTION: CONTEXT-TREE WEIGHTING

### A. Arithmetic Coding

Denote the binary sequence  $x_1x_2 \cdots x_T$  by  $x_1^T$ . Given a coding distribution  $P_c(x_1^T)$  over all binary sequences of length  $T$ , the Elias algorithm (see e.g. Jelinek [1]) generates codewords that satisfy the prefix condition with lengths

$$L(x_1^T) = \lceil \log_2 \frac{1}{P_c(x_1^T)} \rceil + 1 < \log_2 \frac{1}{P_c(x_1^T)} + 2. \quad (1)$$

Implementations of this method are called arithmetic coding methods (e.g. Rissanen [5], Pasco [4]). The codeword length that we obtain in this way is at most two binary digits longer than the length that we desire (i.e. the ideal codeword length  $-\log_2 P_c(x_1^T)$ ). We say that the individual *coding redundancy* is smaller than 2. Therefore universal source coding is mainly concerned with finding good coding distributions.

### B. Krichevski-Trofimov estimator

The actual probability  $\Pr\{X_1^t = x_1^t\}$  of a source sequence  $x_1^t$  for  $t = 1, T$  is denoted as  $P_a(x_1^t)$ . For an independent identically distributed (i.i.d.) binary source with an unknown parameter  $\theta = P_a(1)$  we should use

$$P_e(a, b) = \frac{(a - \frac{1}{2})(a - \frac{3}{2}) \cdots (a - \frac{1}{2})(b - \frac{1}{2})(b - \frac{3}{2}) \cdots (b - \frac{1}{2})}{(a + b)(a + b - 1) \cdots 1} \quad (2)$$

as coding probability for a sequence containing  $a$  zeroes and  $b$  ones. This assignment is called the Krichevski-Trofimov (KT) estimate [2]. Consider a sequence  $x_1^T$  with  $a$  zeroes and  $b$  ones, then from (1) we may conclude that

$$L(x_1^T) < \log_2 \frac{1}{P_e(a, b)} + 2. \quad (3)$$

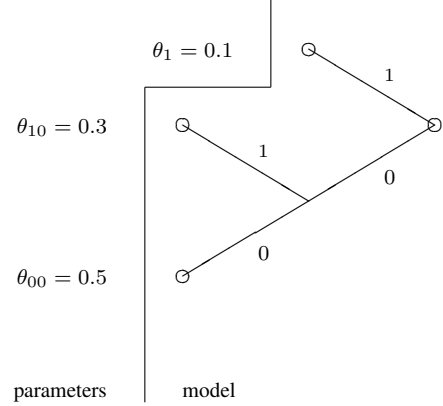


Fig. 1. Model (suffix set) and parameters.

Define the individual redundancy for sequence  $x_1^T$  as

$$\rho(x_1^T) \triangleq L(x_1^T) - \log_2 \frac{1}{P_a(x_1^T)}, \quad (4)$$

then this redundancy for a sequence  $x_1^T$  with  $a$  zeroes and  $b$  ones satisfies

$$\begin{aligned} \rho(x_1^T) &< \log_2 \frac{1}{P_e(a, b)} + 2 - \log_2 \frac{1}{(1 - \theta)^a \theta^b} \\ &= \log_2 \frac{(1 - \theta)^a \theta^b}{P_e(a, b)} + 2 \leq \frac{1}{2} \log_2 T + 3, \end{aligned} \quad (5)$$

where we used lemma 1 from [9] to upper bound the  $\log_2(P_a/P_e)$ -term. This term, called the *parameter redundancy*, is never larger than  $\frac{1}{2} \log_2(a + b) + 1$ . Hence the individual redundancy is not larger than  $\frac{1}{2} \log_2 T + 3$  for all  $x_1^T$  and all  $\theta \in [0, 1]$ . Therefore this estimator is asymptotically optimal (see Rissanen [6]).

### C. Tree Sources

Consider figure 1. For a *tree source* the probability  $P_a(X_t = 1 | \cdots, x_{t-2}, x_{t-1})$  is determined by starting in the root  $\lambda$  of the tree and moving along the path  $x_{t-1}, x_{t-2}, \cdots$  until a leaf of the tree is reached. In this leaf  $s$  we find the desired probability (parameter)  $\theta_s$ . The suffix set or tree  $\mathcal{S}$ , containing the paths to all leaves, is called the *model* of the source.

For the source in figure 1 the actual (conditional) probability of the source generating the sequence 01101 given the past

symbols  $\dots 010$  is:

$$\begin{aligned} P_a(01101|\dots 010) &= (1 - \theta_{10})\theta_{00}\theta_1(1 - \theta_1)\theta_{10} \\ &= 0.00945. \end{aligned} \quad (6)$$

#### D. Unknown Parameters, Known Model

The source model (tree)  $\mathcal{S}$  partitions the source sequence in i.i.d. sub-sequences, one for each leaf  $s \in \mathcal{S}$ . If the parameters of the source are unknown we can use the KT-estimator for each of these sub-sequences. E.g. for  $\mathcal{S} = \{00, 10, 1\}$  we get:

$$P_e(x_1^T|\mathcal{S}) = P_e(a_{00}, b_{00}) \cdot P_e(a_{10}, b_{10}) \cdot P_e(a_1, b_1), \quad (7)$$

where  $a_s$  is the number of zeroes in the subsequence of  $x_1^T$  corresponding to leaf  $s$ , and  $b_s$  the number of ones in this subsequence. In general we obtain for the estimated probabilities for tree-model  $\mathcal{S}$ :

$$P_e(x_1^T|\mathcal{S}) = \prod_{s \in \mathcal{S}} P_e(a_s, b_s). \quad (8)$$

If we use this probability estimate as coding probability, we obtain for the (parameter plus coding) redundancy

$$\begin{aligned} \rho(x_1^T) &< \log_2 \frac{1}{P_e(x_1^T|\mathcal{S})} + 2 - \log_2 \frac{1}{P_a(x_1^T)} \\ &\leq \left( \frac{|\mathcal{S}|}{2} \log_2 \frac{T}{|\mathcal{S}|} + |\mathcal{S}| \right) + 2, \end{aligned} \quad (9)$$

for  $T \geq |\mathcal{S}|$ . Note that the second inequality follows from the convexity of the  $\log_2(\cdot)$ . Moreover  $\rho(x_1^T) < T + 2$  for  $T < |\mathcal{S}|$ .

#### E. Weighting

Consider two sources. For the first source we should use coding distribution  $P_c^1(x_1^T)$  to obtain a small redundancy, for the second source we should use distribution  $P_c^2(x_1^T)$ . If we need a single code that is good for both sources then

$$P_w(x_1^T) = \frac{P_c^1(x_1^T) + P_c^2(x_1^T)}{2} \quad (10)$$

would be a good coding distribution for this code. It leads to codeword length

$$\begin{aligned} L_w(x_1^T) &< \log_2 \frac{2}{P_c^1(x_1^T) + P_c^2(x_1^T)} + 2 \\ &\leq \log_2 \frac{1}{P_c^i(x_1^T)} + 3, \text{ for } i = 1, 2, \end{aligned} \quad (11)$$

and we loose at most one binary digit with this weighting technique!

#### F. Unknown Model

Suppose that the actual source model  $\mathcal{S}$  is unknown, but that its depth is not larger than  $D$ . A *context* is a string of binary symbols. Note that to each context  $s$ , there corresponds a substring of  $x_1 \dots x_T$  of symbols that are produced by the source following this context  $s$ . Let  $a_s$  be the number of zeroes in this subsequence and  $b_s$  the number of ones. The structure that contains a node for all contexts  $s$  having depth not larger than  $D$  is called the *context tree*  $\mathcal{T}_D$ . A good probability

estimate for the subsequence corresponding to a context (node)  $s$  at depth  $D$  is  $P_w^s = P_e(a_s, b_s)$ . Now let the depth  $d$  of some node  $s$  be less than  $D$  and assume that we already have good probability estimates for subsequences corresponding to nodes  $0s$  and  $1s$  at depth  $d+1$ . Denote these probability estimates by  $P_w^{0s}$  and  $P_w^{1s}$  respectively. For the subsequence corresponding to  $s$  we then have two alternatives. We can use the KT-estimate  $P_e(a_s, b_s)$  for the entire subsequence corresponding to  $s$ , or we can split up this subsequence in two sub-subsequences and use the product  $P_w^{0s} P_w^{1s}$  of the probabilities  $P_w^{0s}$  and  $P_w^{1s}$  as estimate. If we weight these two alternatives we obtain the weighted probability

$$P_w^s = \begin{cases} \frac{1}{2} P_e(a_s, b_s) + \frac{1}{2} P_w^{0s} P_w^{1s} & \text{if depth}(s) < D, \\ P_e(a_s, b_s) & \text{otherwise.} \end{cases} \quad (12)$$

The weighted probability  $P_w^\lambda$  in the root of the context tree can now be used as coding probability for the entire sequence  $x_1^T$ . The method is called the context-tree weighting (CTW) method. Important is that the weighted probability realized by CTW satisfies

$$\begin{aligned} P_w^\lambda &= \sum_{\mathcal{S}} 2^{-\Gamma_D(\mathcal{S})} \cdot \prod_{s \in \mathcal{S}} P_e(a_s, b_s), \\ &\geq 2^{-\Gamma_D(\mathcal{S}_a)} \cdot \prod_{s \in \mathcal{S}_a} P_e(a_s, b_s), \end{aligned} \quad (13)$$

where the summation is over all tree models that fit in the context tree  $\mathcal{T}_D$ , see Lemma 2 in [9]. The cost of model  $\mathcal{S}$  is defined as

$$\Gamma_D(\mathcal{S}) \triangleq 2|\mathcal{S}| - 1 - |\{s \in \mathcal{S}, \text{depth}(s) = D\}|, \quad (14)$$

and  $\mathcal{S}_a$  is the actual model.

#### G. Performance

The individual redundancy  $\rho(x_1^T)$  relative to the actual source for sequence  $x_1^T$  can be upper bounded by

$$\begin{aligned} \rho(x_1^T) &= L_w(x_1^T) - \log_2 \frac{1}{P_a(x_1^T)} \\ &< \Gamma_D(\mathcal{S}_a) + \frac{|\mathcal{S}_a|}{2} \log_2 \frac{T}{|\mathcal{S}_a|} + |\mathcal{S}_a| + 2, \end{aligned} \quad (15)$$

for  $T \geq |\mathcal{S}_a|$ . Moreover  $\rho(x_1^T) < \Gamma_D(\mathcal{S}_a) + T + 2$  for  $T < |\mathcal{S}_a|$ . The three terms in bound (15) are the cost of specifying the model i.e.  $\Gamma_D(\mathcal{S}_a)$ , the cost of specifying the parameters which is  $\frac{|\mathcal{S}_a|}{2} \log_2 \frac{T}{|\mathcal{S}_a|} + |\mathcal{S}_a|$  and the loss of 2 binary digits due to arithmetic coding.

Observe that bound (15) also holds for the redundancy relative to *any other source tree model*  $\mathcal{S}$  with depth  $\leq D$ .

## II. RATIOS OF PROBABILITIES: BETAS

To reduce the complexity we can store in a node instead of the estimated and weighted block probability a *probability ratio*. This idea was proposed in [11]. Consider an internal node  $s$  in the context tree  $\mathcal{T}_D$  and the corresponding *conditional* weighted probability  $P_w^s(X_t = 1|x_1^{t-1}, x_{1-D}^0)$ . Assuming that

0s (and not 1s) is a suffix of the context  $x_{1-D}^0, x_1^{t-1}$  of  $x_t$ , we obtain for this probability that

$$\begin{aligned} P_w^s(X_t = 1 | x_1^{t-1}, x_{1-D}^0) \\ &= \frac{P_e^s(x_1^{t-1}, X_t = 1) + P_w^{0s}(x_1^{t-1}, X_t = 1)P_w^{1s}(x_1^{t-1})}{P_e^s(x_1^{t-1}) + P_w^{0s}(x_1^{t-1})P_w^{1s}(x_1^{t-1})} \\ &= \frac{\beta^s(x_1^{t-1})P_e^s(X_t = 1 | x_1^{t-1}) + P_w^{0s}(X_t = 1 | x_1^{t-1})}{\beta^s(x_1^{t-1}) + 1} \end{aligned} \quad (16)$$

where

$$\beta^s(x_1^{t-1}) = \frac{P_e^s(x_1^{t-1})}{P_w^{0s}(x_1^{t-1})P_w^{1s}(x_1^{t-1})}. \quad (17)$$

Here we used in the first equality the main CTW-definition (12) and the fact that  $P_w^{1s}(x_1^{t-1}, X_t = 1) = P_w^{1s}(x_1^{t-1})$  since 1s is not a suffix of the context  $x_{1-D}^0, x_1^{t-1}$ . For simplicity we have omitted  $x_{1-D}^0$  in all conditions.

Assuming that in node  $s$  the counts  $a_s(x_1^{t-1})$  and  $b_s(x_1^{t-1})$  are stored, as well as the ratio  $\beta^s(x_1^{t-1})$ , leads to the following sequence of operations:

- 1) Node 0s delivers the conditional weighted probability  $P_w^{0s}(X_t = 1 | x_1^{t-1})$  to node  $s$ .
- 2) A conditional estimated probability is determined as suggested by Krichevsky and Trofimov [2], i.e.:

$$P_e^s(X_t = 1 | x_1^{t-1}) = \frac{b_s(x_1^{t-1}) + 1/2}{a_s(x_1^{t-1}) + b_s(x_1^{t-1}) + 1}. \quad (18)$$

The block-version of this KT-estimator appears in (2).

- 3) Now the outgoing conditional weighted probability can be computed as described in (16).
- 4) The ratio  $\beta^s(\cdot)$  is then updated with symbol  $x_t$ . This is done as described below:

$$\beta^s(x_1^{t-1}, x_t) = \beta^s(x_1^{t-1}) \cdot \frac{P_e^s(X_t = x_t | x_1^{t-1})}{P_w^{0s}(X_t = x_t | x_1^{t-1})}. \quad (19)$$

- 5) Finally, depending on the value  $x_t$ , either count  $a_s(x_1^{t-1})$  or  $b_s(x_1^{t-1})$  is incremented.

We see that inside the node  $s$  there is a *switch* that controls the mixture between the incoming conditional weighted probability  $P_w^{0s}(X_t = 1 | x_1^{t-1})$  and the (internal) ratio  $P_e^s(X_t = 1 | x_1^{t-1})$ . The mixture is determined by the ratio  $\beta^s(x_1^{t-1})$ . For large  $\beta^s(x_1^{t-1})$  the outgoing conditional probability is roughly  $P_e^s(X_t = 1 | x_1^{t-1})$ , for small  $\beta^s(x_1^{t-1})$  it is approximately equal to  $P_w^{0s}(X_t = 1 | x_1^{t-1})$ . If  $s$  is a leaf of  $\mathcal{T}_D$  the outgoing conditional weighted probability is simply  $P_e^s(X_t = 1 | x_1^{t-1})$ , i.e. the internal one.

A storage complexity reduction is obtained since estimated and weighted block probabilities decrease as the sequence length  $T$  increases, while the ratio  $\beta^s$  corresponds to two different coding alternatives for the subsequence in the node  $s$  and is therefore closer to one. Moreover Eq. (16) shows that in practice the performance does not depend on how large and how small  $\beta^s$  really can become as long as it is large enough and small enough.

### III. ON A LINEAR COMBINATION

We can use equation (16) to express the conditional weighted probability of the root as a linear combination of the estimated probabilities of the nodes along the context path  $x_{t-D}x_{t-D+1} \cdots x_{t-1}$ . This results in

$$P_w^\lambda(x_1^{t-1}) = \sum_{d=0, D} \mu^{s_d}(x_1^{t-1}) P_e^s(X_t = 1 | x_1^{t-1}) \quad (20)$$

where  $s_0 \triangleq \lambda$  and  $s_d \triangleq x_{t-d} \cdots x_{t-1}$  for  $d = 1, \dots, D$ , and

$$\mu^{s_d}(x_1^{t-1}) = \frac{\beta^{s_d}(x_1^{t-1})}{\beta^{s_d}(x_1^{t-1}) + 1} \prod_{i=0, d-1} \frac{1}{\beta^{s_i}(x_1^{t-1}) + 1}, \quad (21)$$

for  $d = 0, 1, \dots, D-1$  and

$$\mu^{s_D}(x_1^{t-1}) = \prod_{i=0, D-1} \frac{1}{\beta^{s_i}(x_1^{t-1}) + 1}. \quad (22)$$

If we observe that  $\mu^{s_d}(x_1^{t-1}) \geq 0$  for  $d = 0, 1, \dots, D$  and

$$\sum_{d=0, D} \mu^{s_d}(x_1^{t-1}) = 1, \quad (23)$$

we may conclude that (20) is actually a convex combination.

### IV. COMPUTING A POSTERIORI MODEL PROBABILITIES

Consider a sub-tree model  $\mathcal{S}_s$  (a proper and complete set of strings all having a common suffix  $s$ ), rooted in node  $s$  of  $\mathcal{T}_D$  and fitting in the context tree  $\mathcal{T}_D$ . Then we define the "conditional" probability of the sub-tree  $\mathcal{S}_s$  given  $x_1^T$  as

$$Q_w^s(\mathcal{S}_s) \triangleq \frac{2^{-\Gamma_D(\mathcal{S}_s)} \prod_{s \in \mathcal{S}_s} P_e(a_s, b_s)}{P_w^s}, \quad (24)$$

where the cost of sub-model  $\mathcal{S}_s$  is defined as

$$\Gamma_D(\mathcal{S}_s) \triangleq 2|\mathcal{S}_s| - 1 - |\{s \in \mathcal{S}_s, \text{depth}(s) = D\}|. \quad (25)$$

It makes sense to call this probability a conditional probability since the denominator in (24) can be expressed as

$$P_w^s = \sum_{\mathcal{S}_s} 2^{-\Gamma_D(\mathcal{S}_s)} \prod_{s \in \mathcal{S}_s} P_e(a_s, b_s), \quad (26)$$

and

$$\sum_{\mathcal{S}_s} 2^{-\Gamma_D(\mathcal{S}_s)} = 1, \quad (27)$$

where the summations are over all sub-models rooted in  $s$  having no leaves at depth deeper than  $D$ . This is Lemma 2 in [9].

Now if  $|\mathcal{S}_s| > 1$ , note that the node  $s$  can not be at level  $D$  then, we can split up the sub-model  $\mathcal{S}_s$  into a sub-model  $\mathcal{S}_{0s}$  and a sub-model  $\mathcal{S}_{1s}$  and we obtain for the conditional probability

$$\begin{aligned} Q_w^s(\mathcal{S}_s) &= \frac{2^{-\Gamma_D(\mathcal{S}_{0s})} \prod_{s \in \mathcal{S}_{0s}} P_e(a_s, b_s)}{P_w^{0s}} \\ &\cdot \frac{2^{-\Gamma_D(\mathcal{S}_{1s})} \prod_{s \in \mathcal{S}_{1s}} P_e(a_s, b_s)}{P_w^{1s}} \\ &\cdot \frac{P_w^{0s} P_w^{1s}}{P_e(a_s, b_s) + P_w^{0s} P_w^{1s}} \\ &= Q_w^{0s}(\mathcal{S}_{0s}) Q_w^{1s}(\mathcal{S}_{1s}) \frac{1}{\beta_s + 1}. \end{aligned} \quad (28)$$

When the sub-model  $\mathcal{S}_s$  contains only one leaf-node  $s$ , not at depth  $D$ , then

$$Q_w^s(\mathcal{S}_s) = \frac{P_e(a_s, b_s)}{P_e(a_s, b_s) + P_w^{0s} P_w^{1s}} = \frac{\beta_s}{\beta_s + 1}. \quad (29)$$

If sub-model  $\mathcal{S}_s$  consists only of a single leaf-node  $s$  at level  $D$  then

$$Q_w^s(\mathcal{S}_s) = 1. \quad (30)$$

Summarizing the three considered cases we can write

$$Q_w^s(\mathcal{S}_s) = \begin{cases} Q_w^{0s}(\mathcal{S}_{0s}) Q_w^{1s}(\mathcal{S}_{1s}) \frac{1}{\beta_s + 1} & \text{if } |\mathcal{S}_s| > 1, \\ \frac{\beta_s}{\beta_s + 1} & \text{if } \text{depth}(s) < D \text{ for } |\mathcal{S}_s| = 1, \\ 1 & \text{if } \text{depth}(s) = D \text{ for } |\mathcal{S}_s| = 1. \end{cases} \quad (31)$$

If we take

$$P(\mathcal{S}) \triangleq 2^{-\Gamma_D(\mathcal{S})} \quad (32)$$

as the *a priori probability* of model  $\mathcal{S}$ , then we can write for the *a posteriori probability* of model  $\mathcal{S}$  after having observed  $x_1^T$  that

$$P_w(\mathcal{S}|x_1^T) = \frac{2^{-\Gamma_D(\mathcal{S})} \prod_{s \in \mathcal{S}} P_e(a_s, b_s)}{P_w^\lambda} = Q_w^\lambda(\mathcal{S}), \quad (33)$$

where the last equality follows from (24). Recursive expression (31) can now be used to determine the a posteriori probability  $Q_w^\lambda(\mathcal{S})$  of a model  $\mathcal{S}$  from the  $\beta$ 's in the context tree. We just have to form a product which consists of a factor  $1/(\beta_{s'} + 1)$  for each internal node  $s'$  of the model  $\mathcal{S}$  and a factor  $\beta_{s''}/(\beta_{s''} + 1)$  for each leaf  $s''$  of the model  $\mathcal{S}$  not at level  $D$ .

## V. INTRODUCTION: CONTEXT-TREE MAXIMIZING

### A. Two-pass methods

CTW is a *one-pass method*. The source sequence  $x_1^T$  is processed in a sequential way, i.e. the first source symbol  $x_1$  is observed, some first code symbols may be produced, the second symbol  $x_2$  is observed, more code symbols may be produced, etc. . In a *two-pass system* the entire source sequence  $x_1^T$  is observed first. Only after that a codeword is constructed. Consider the following *two-pass method*:

- 1) After observing  $x_1^T$  determine the “best model”  $\hat{\mathcal{S}}$  matching to  $x_1^T$ .
- 2) Encode this model  $\hat{\mathcal{S}}$ .
- 3) Encode the sequence  $x_1^T$  given this model  $\hat{\mathcal{S}}$ .

Some questions that arise now are: (a) What is the best model  $\hat{\mathcal{S}}$ ? How can it be determined efficiently? (b) How do we encode the best model  $\hat{\mathcal{S}}$  and sequence  $x_1^T$  given model  $\hat{\mathcal{S}}$ ?

### B. The context-tree maximizing algorithm

When the source produces sequence  $x_1^T$  and model  $\mathcal{S}$  is chosen as the best model, the resulting coding probability<sup>1</sup> for the two-pass case, is

$$2^{-\Gamma_D(\mathcal{S})} \cdot \prod_{s \in \mathcal{S}} P_e(a_s, b_s). \quad (34)$$

<sup>1</sup>It satisfies  $\sum_{x_1^T} P_m^\lambda(x_1^T) < 1$  however.

Here the first factor is the number of bits needed to specify the model  $\mathcal{S}$  in a recursive way (i.e. the natural code mentioned in [9]) and the second factor is the coding probability of the sequence  $x_1^T$  given the model  $\mathcal{S}$ , see (8). The context-tree maximizing (CTM) method, first mentioned in [8] but also proposed by Nohre in his Ph.D. thesis [3]), finds the model maximizing (34) recursively, using a context tree, by taking

$$P_m^s = \begin{cases} \max[\frac{1}{2}P_e(a_s, b_s), \frac{1}{2}P_m^{0s}P_m^{1s}] & \text{if } \text{depth}(s) < D, \\ P_e(a_s, b_s) & \text{otherwise.} \end{cases} \quad (35)$$

We assumed that the entire sequence  $x_1^T$  was processed into the context tree. We finally will find the best model  $\mathcal{S}$  by tracking the maximization procedure, starting in the *root*  $\lambda$  of the context tree. If in a node  $s$  in the context tree  $P_e(a_s, b_s) \geq P_m^{0s}P_m^{1s}$  then  $s$  is a leaf of the best tree  $\hat{\mathcal{S}}$  and we do not have to investigate the sub-tree rooted in  $s$  any further. Otherwise  $s$  is an internal node of the best model and we have to check the nodes  $0s$  and  $1s$ . Note that

$$P_m^\lambda = \max_{\mathcal{S}} 2^{-\Gamma_D(\mathcal{S})} \prod_{s \in \mathcal{S}} P_e(a_s, b_s), \quad (36)$$

and denote that model maximizing this expression by  $\hat{\mathcal{S}}$ .

### C. Performance

The coding probability for context-tree maximizing satisfies

$$P_m^\lambda = 2^{-\Gamma_D(\hat{\mathcal{S}})} \prod_{s \in \hat{\mathcal{S}}} P_e(a_s, b_s) \geq 2^{-\Gamma_D(\mathcal{S}_a)} \prod_{s \in \mathcal{S}_a} P_e(a_s, b_s), \quad (37)$$

just like  $P_w^\lambda$ , see (13), and therefore maximizing, just like weighting, leads to the redundancy bound (15). Observe that this bound holds for any model, not only for  $\mathcal{S}_a$ .

## VI. FINDING THE MAXIMUM A POSTERIORI MODEL

Observe first that the context-tree maximizing method yields the maximum a posteriori (MAP) tree model given the observed sequence  $x_1^T$ . This observation can be found in [7]. In [7] also the method for computing a posteriori model probabilities, that was described in section IV, was proposed. It is a bit strange that on one hand a posteriori model probabilities can be computed from the  $\beta$ 's in a context tree while on the other hand to determine the MAP tree-model we need the context-tree maximizing method. Therefore we want to find a method that determines the MAP-model based on the  $\beta$ 's in the weighted context-tree. First we determine the probability of the MAP sub-model corresponding to a node  $s$  at depth  $< D$ . For such a node we can write

$$\begin{aligned} \max_{\mathcal{S}_s} Q_w^s(\mathcal{S}_s) \\ = \max[\frac{1}{\beta_s + 1} \max_{\mathcal{S}_{0s}} Q_w^{0s}(\mathcal{S}_{0s}) \max_{\mathcal{S}_{1s}} Q_w^{1s}(\mathcal{S}_{1s}), \frac{\beta_s}{\beta_s + 1}]. \end{aligned} \quad (38)$$

Here the last term corresponds to the sub-model which has only a single leaf-node at  $s$ . The first term correspond to all larger sub-models.

For a node at depth  $D$  only the one-leaf sub-model plays a role and

$$\max_{\mathcal{S}_s} Q_w^s(\mathcal{S}_s) = 1. \quad (39)$$

If we now define for all nodes  $s \in \mathcal{T}_D$  the MAP sub-model probability

$$Q_{mw}^s \triangleq \max_{\mathcal{S}_s} Q_w^s(\mathcal{S}_s), \quad (40)$$

then the following recursive equation holds:

$$Q_{mw}^s = \begin{cases} \max[Q_{mw}^{0s} Q_{mw}^{1s} \frac{1}{\beta_s+1}, \frac{\beta_s}{\beta_s+1}] & \text{if depth}(s) < D, \\ 1 & \text{if depth}(s) = D. \end{cases} \quad (41)$$

Now in the root  $\lambda$  of the context tree we find the maximum a posteriori model probability  $Q_{mw}^\lambda$ . Tracking the procedure starting in the root of the context tree yields the MAP-model.

## VII. DIFFERENCE BETWEEN CTW AND CTM

Let  $\hat{\mathcal{S}}$  be the MAP model, then

$$1 \geq \frac{2^{-\Gamma_D(\hat{\mathcal{S}})} \prod_{s \in \hat{\mathcal{S}}} P_e(a_s, b_s)}{P_w^\lambda} = Q_w^\lambda(\hat{\mathcal{S}}) = Q_{mw}^\lambda. \quad (42)$$

For the difference in codeword lengths for CTW and CTM we first can write

$$\begin{aligned} L_w(x_1^T) - L_m(x_1^T) &= \lceil \log_2 \frac{1}{P_w^\lambda(x_1^T)} \rceil + 1 - \lceil \log_2 \frac{1}{P_m^\lambda(x_1^T)} \rceil - 1 \\ &\leq 0. \end{aligned} \quad (43)$$

However we can also show that

$$\begin{aligned} L_w(x_1^T) - L_m(x_1^T) &< \log_2 \frac{1}{P_w^\lambda(x_1^T)} + 2 - \log_2 \frac{1}{P_m^\lambda(x_1^T)} - 1 \\ &= \log_2 \frac{2^{-\Gamma_D(\hat{\mathcal{S}})} \prod_{s \in \hat{\mathcal{S}}} P_e(a_s, b_s)}{P_w^\lambda} + 1 \\ &= \log_2 Q_{mw}^\lambda + 1, \end{aligned} \quad (44)$$

and similarly

$$L_w(x_1^T) - L_m(x_1^T) > \log_2 Q_{mw}^\lambda - 1. \quad (45)$$

Note that only for  $Q_{mw}^\lambda \approx 1$  we obtain that  $L_m(x_1^T) \leq L_w(x_1^T) + 1$  and CTM yields roughly the same performance as CTW. In [10] it was shown that for tree sources that fit in the context tree  $\mathcal{T}_D$  MAP probability  $Q_{mw}^\lambda \rightarrow 1$  for asymptotically large sequence length  $T$ . When  $Q_{mw}^\lambda$  is not close to one (no convergence)  $L_m(x_1^T)$  is significantly larger than  $L_w(x_1^T)$ .

## VIII. A POSTERIORI NODE PROBABILITIES

We can define the a posteriori node probability of a node  $s \in \mathcal{T}_D$  as

$$Q_w(s) \triangleq \sum_{\mathcal{S}: s \in \mathcal{S}} Q_w^\lambda(\mathcal{S}), \quad (46)$$

where the summation is over all models  $\mathcal{S}$  that contain leaf  $s$ .

It now can be shown that for all  $s \in \mathcal{T}_D$

$$\mu^s = Q_w(s), \quad (47)$$

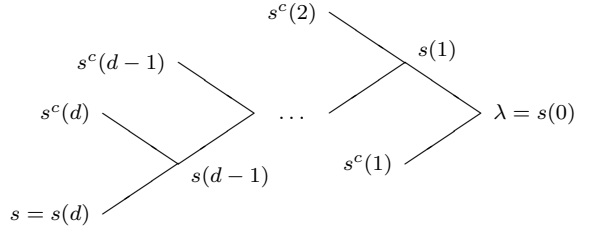


Fig. 2. A leaf at depth  $d$  and the path leading to this leaf.

where  $\mu_s$  is as in (21) and (22).

For a proof of this statement we refer to figure 2. Let node  $s$  have depth  $d$  and assume that  $s = u_d, u_{d-1}, \dots, u_1$ , a binary string. Then define  $s(i) \triangleq u_i, u_{i-1}, \dots, u_1$  and  $s^c(i) \triangleq 1 - u_i, u_{i-1}, \dots, u_1$  for  $i = 1, \dots, d$  and  $s(0) \triangleq \lambda$ . The crucial part of the proof is (31). Note that the nodes  $s(i), i = 0, 1, \dots, d-1$  are inner nodes of all models that have a leaf  $s$ . These nodes give rise to the factors in (21) and (22). Moreover we use that

$$\sum_{\mathcal{S}_{s^c(i)}} Q_w^{s^c(i)}(\mathcal{S}_{s^c(i)}) = 1. \quad (48)$$

for all  $i = 1, 2, \dots, d$ .

## IX. CONCLUDING REMARKS

We have presented here several new results related to  $\beta$ 's. The results are all based on uniform weighting, i.e. using coefficients  $\frac{1}{2}, \frac{1}{2}$  if we weight two alternatives. Our results carry over to the case where arbitrary coefficients are used. Note that a uniform distribution over all tree-models can be achieved by a special kind of non-uniform weighting (see [9]).

## 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] R. Nohre, *Some Topics in Descriptive Complexity*, Ph.D. thesis, Linköping University, Sweden, 1994.
- [4] R. Pasco, *Source Coding Algorithms for Fast Data Compression*, Ph.D. thesis, Stanford University, 1976.
- [5] J. Rissanen, "Generalized Kraft Inequality and Arithmetic Coding," *IBM J. Res. Devel.*, vol. 20, p. 198, 1976.
- [6] J. Rissanen, "Universal Coding, Information, Prediction, and Estimation," *IEEE Inform. Theory*, vol. IT-30, pp. 629 - 636, July 1984.
- [7] F.M.J. Willems, A. Nowbahkt-irani, and P.A.J. Volf, "Maximum A Posteriori Tree Models," *Proc. 4th Int. ITG Conf. Source and Channel Coding*, Berlin, Germany, pp. 335 - 340, February 28-30, 2002.
- [8] F.M.J. Willems, Y.M. Shtarkov and T.J.J. Tjalkens, "Context Weighting: General Finite Context Sources," *Proc. 114th Symp. on Inform. Theory in the Benelux*, pp. 120 - 127, Veldhoven, May 17 & 18, 1993.
- [9] F.M.J. Willems, Y.M. Shtarkov and T.J.J. Tjalkens, "The Context-Tree Weighting Method: Basic Properties," *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 653 - 664, May 1995.
- [10] F.M.J. Willems, Y.M. Shtarkov and T.J.J. Tjalkens, "Context-Tree Maximizing," *2000 Conf. on Inform. Sciences and Syst.*, Princeton Univ., Princeton, NJ, TP6.7 - TP6.12, March 15 - 17, 2000.
- [11] F.M.J. Willems and T.J.J. Tjalkens, "Reducing complexity of the context-tree weighting method," *Proc. IEEE International Symposium on Information Theory*, p. 347, Cambridge, Mass., August 16 - 21, 1998.