Context quantization by minimum adaptive code length

Forchhammer, Søren; Wu, Xiaolin

Published in:
IEEE International Symposium on Information Theory, 2007. ISIT 2007.

Link to article, DOI:
10.1109/ISIT.2007.4557234

Publication date:
2007

Document Version
Publisher's PDF, also known as Version of record

Context Quantization by Minimum Adaptive Code Length

Søren Forchhammer
COM Department, B. 343
Technical University of Denmark
DK-2800 Lyngby, Denmark
sf@com.dtu.dk

Xiaolin Wu
Dept. Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, Canada L8G 4M2
xwu@mail.ece.mcmaster.ca

***Abstract*— Context quantization is a technique to deal with the issue of context dilution in high-order conditional entropy coding. We investigate the problem of context quantizer design under the criterion of minimum adaptive code length. A property of such context quantizers is derived for binary symbols. A fast context quantizer design algorithm for conditioning binary symbols is presented and its complexity analyzed. It is conjectured that this algorithm is optimal. The context quantization is performed in what may be perceived as a probability simplex space rather than in the space of context instances.**

I. Introduction

A key problem in sequential source coding of a discrete random sequence $X_1, X_2, X_3, \cdots$ is estimating the conditional probabilities $P(X_i|X^{i-1})$, where X^{i-1} denotes $X_1, X_2, \cdots, X_{i-1}$, the prefix of X_i. Coding can be implemented sequentially by arithmetic coding based on the estimated conditional probabilities. Because of the model cost due to parameter estimation, it is not efficient to directly use a large subset of X^{i-1} as the context used for predicting X_i. But using few contexts may lump together contexts with different statistics. Indeed, given a model class, the order of the model or the number of model parameters needs to be carefully selected so as not to negatively impact the code length. In the literature, this problem is addressed in various ways. Rissanen uses the minimum description length principle in the universal algorithm *Context* [1] to select a variable-order context, where the candidate contexts are tree structured. Universal source codes are optimal asymptotically in the length of the source sequence, relative to some class of models. Algorithm *Context* (in a slightly modified version [2]), and the related *Context Tree Weighting* algorithm [3], were shown to be universal for the class of tree sources.

From a practical point of view, these *Context* algorithms are complex and they do not use prior domain knowledge. In [4] an image coder inspired by Algorithm *Context* utilizing the smoothness of gray-scale images was introduced. Many practical source coders choose *a priori* a coding model with fixed complexity, based on domain knowledge such as correlation structure and typical data length, and estimate only the model parameters. For example, the JBIG [5] bi-level image compression standard employs template coding where the next symbol X_i is encoded using as a modeling context a pre-chosen set of d past symbols $C_i = (X_{i-t_1}, X_{i-t_2}, \ldots, X_{i-t_d})$.

Estimating $P(X_i|C_i)$ directly with past sample statistics may be inefficient due to the model cost if the template of the context is large and/or the symbol alphabet is large. To alleviate this problem, the state-of-the-art lossless image compression algorithm CALIC [6], the JPEG-LS [7], [8] image compression standard, and the JPEG 2000 entropy coding algorithm EBCOT [9] quantize the modeling context into a relatively small number of conditioning states, and estimate $P(X_i|Q(C_i))$ instead of $P(X_i|C_i)$, where Q is a context quantizer. Central to the performance of this approach is the design of the context quantizer Q. From this point of view Algorithm *Context* applies an adaptive tree structured context quantization.

Heuristic algorithms for designing context quantizers under the criterion of minimum code length [6], [10] have been proposed. These algorithms are not strictly universal. By using domain knowledge, they have produced some of the best performing signal compression algorithms. Using training sequences to design the context quantizer is one form of prior domain knowledge. In this case even relatively large contexts may be used keeping the model cost under control by quantizing the contexts, lumping together contexts with similar model parameters, without imposing a (tree) structure on the quantization.

The main problem we consider is the design of a context quantizer (CQ) Q, with M states, that maps the $K-$dimensional context, C_i into one of M conditioning states. This has been studied with the design criterion of minimum conditional entropy [11]. But the optimum number of states M was left an open issue (or predefined term). We consider CQ using adaptive code length as the objective function to be minimized [13]. This incurs a model cost and thus leads to an optimum value of M in the chosen setting. The $K-$dimensional context space is fixed and given prior to the context quantization. The CQ design problem may be viewed as one of vector quantization with achievable code length being the objective function. Consequently, the quantization is performed in a space related to the objective function instead of the $K-$dimensional space of context instances.

Section II gives a brief introduction of context quantization. In Section III, the problem of optimal CQ design for minimum

1-4244-1429-6/07/$25.00 ©2007 IEEE

adaptive code length is formulated, and some properties of optimal CQ for binary symbols are discussed. In Section IV, a fast CQ design algorithm is presented and its complexity determined. It is conjectured that the algorithm is optimal.

II. Context Quantization

Let Y be a random variable with finite alphabet $\mathcal{Y}$, and let C be a jointly distributed random variable with abstract alphabet $\mathcal{C}$. The random variable C will be the *context* for predicting the random variable Y.

Given a positive integer M, let $Q : \mathcal{C} \to \{1, 2, \ldots, M\}$ be a measurable function of C. The M-ary function Q will be called a *context quantizer* for the context C. The value of the context quantizer $m = Q(\mathbf{c})$ for any given context instance $\mathbf{c} \in \mathcal{C}$ will be called the *conditioning state* of the context.

The quantization cells $A_m = \{\mathbf{c} : Q(\mathbf{c}) = m\}$, $m = 1, \ldots, M$, of a CQ may be quite complex in shape (assuming a topology on $\mathcal{C}$) and may not even be connected.

A context quantizer Q may be used to compress any input sequence $y^I = y_1, \ldots, y_I$ given the corresponding raw context sequence $\mathbf{c}^I = \mathbf{c}_1, \ldots, \mathbf{c}_I$, by mapping each context vector $\mathbf{c}_i$ to a conditioning state m, and then arithmetically coding y_i in this conditioning state based on an adaptive probability estimate. We consider sequential coding of $y_1, \ldots, y_I$. (Often the context $\mathbf{c}_i$ is a function of the causal data, y^{i-1}, but we consider the more general case where the contexts could also include other information.)

We restrict the following development to one binary data sequence y^I and the corresponding context sequence $\mathbf{c}^I$, for which all $\mathbf{c}_i$ are drawn from a finite set. The objective function considered here when optimizing the context quantizer is the adaptive code length.

III. Minimizing Adaptive Code Length

For a binary input sequence y^I given $\mathbf{c}^I$ and a context quantizer $Q(\mathbf{c})$, the adaptive code length and the minimum value thereof are introduced below. Adaptive coding sequentially updates the state-conditional probabilities, $\hat{P}_i(y|Q(\mathbf{c}_i) = m)$ at instance i, during arithmetic coding of the input sequence y^I given $\mathbf{c}^I$. This can be done using for example the causal estimator

$$\hat{P}_i(y|Q(\mathbf{c}_i) = m) = \frac{n_i(y, m) + \delta}{n_i(0, m) + n_i(1, m) + 2\delta}, \qquad (1)$$

where $\delta \geq 0$ is a parameter of the estimator and

$$n_i(y, m) = |\{t < i : y_t = y, Q(\mathbf{c}_t) = m\}|$$

is the number of occurrences of y in conditioning state m in the input sequence y^{i-1} prior to y_i. It is straightforward to show [12] that with the estimator (1), the adaptive code length in conditioning state m, namely

$$\ell_m = -\sum_{i=1}^{I} 1\{Q(\mathbf{c}_i) = m\} \log \hat{P}_i(y_i|Q(\mathbf{c}_i)) = m),$$

where $1\{Q(\mathbf{c}_i) = m\}$ is 1 if $Q(\mathbf{c}_i) = m$ and 0 otherwise, does not depend on the *order of appearance* of 0's and 1's but rather only on the *numbers* of 0's and 1's in the conditioning state. Indeed, denoting the numbers of 0's and 1's in conditioning state m as $n_0(m)$ and $n_1(m)$, ℓ_m equals

$$\ell(n_0(m), n_1(m)) = \sum_{j=0}^{n_0+n_1-1} \log(2\delta + j) - \sum_{j=0}^{n_0-1} \log(\delta + j) - \sum_{j=0}^{n_1-1} \log(\delta + j), \qquad (2)$$

where $\delta > 0$ and $\sum_{j=0}^{n-1} \log(\delta + j) \equiv 0$ for $n = 0$.

Definition. The adaptive code length, $L(y^I|Q(\mathbf{c}^I))$, for a binary input sequence y^I given $\mathbf{c}^I$, Q, M and δ is defined by

$$L(y^I|Q(\mathbf{c}^I)) = \sum_{m=1}^{M} \ell(n_0(m), n_1(m)), \qquad (3)$$

where $\ell(n_0, n_1)$ is given by (2).

Our objective is to minimize $L(y^I|Q(\mathbf{c}^I))$ in (3) over all possible quantizers Q of M cells and over all possible non-negative integers M. The resulting context quantizer is called *minimum adaptive code length context quantizer*.

Thus by definition a minimum adaptive code length context quantizer for a given binary input sequence provides a lower bound on the adaptive code length (3) for the data given the contexts $\mathbf{c}_i$ and the chosen value of δ.

As the adaptive code length only depends on the occurrence counts and not the order of 0's and 1's within a conditioning state, the optimal context quantizer is also only a function of the counts. Some properties of the structure of optimal quantization cells are given below, notably for contexts having identical counts or identical ratios of counts.

Lemma III.1. Given integer occurrence counts, $n_0, n_1, a_0, a_1 \geq 0, a_0 + a_1 \neq 0$, and real valued $d_a > d_b \geq 0$

$$\ell(n_0+(d_a+1)a_0, n_1+(d_a+1)a_1) - \ell(n_0+d_a a_0, n_1+d_a a_1) \leq \ell(n_0+(d_b+1)a_0, n_1+(d_b+1)a_1) - \ell(n_0+d_b a_0, n_1+d_b a_1)$$

Proof. See Appendix 1.

Given the occurrence counts in a context, the *occurrence count ratio* is defined by $n_0/(n_0 + n_1)$, i.e. the same as by the expression in (1) for $\delta = 0$. Let $\bar{Q}(\mathbf{c})$ denote a context quantizer for which any pair of contexts having identical occurrence count ratios are mapped to the same context quantizer cell, A_m.

Lemma III.2. Consider a given binary input sequence y^I and given $\mathbf{c}^I$. Restricting the class of context quantizers to $\bar{Q}(\mathbf{c})$ does not change the minimum adaptive code length, in comparison with $Q(\mathbf{c})$ as

$$\min L(y^I|\bar{Q}(\mathbf{c}^I)) = \min L(y^I|Q(\mathbf{c}^I)).$$

Proof. Consider the case that more than one context $\mathbf{c}$ has the counts (a_0, a_1). Consider a mapping $Q(\mathbf{c})$ in which the contexts with identical counts (a_0, a_1) are mapped to two or more different cells. Choose two of these cells which have i and j contexts having identical counts (a_0, a_1), respectively. Let the total counts in the cells give the adaptive code

lengths $(\ell(n_0+ia_0, n_1+ia_1))$ and $(\ell(m_0+ja_0, m_1+ja_1)$, respectively. Define $\Delta\ell(\mathbf{n}+i\mathbf{a}) = \ell(n_0+ia_0, n_1+ia_1) - \ell(n_0+(i-1)a_0, n_1+(i-1)a_1))$, $i>0$. This expresses the incremental code length, coding the occurrences (a_0, a_1).

Assume without loss of generality that $\Delta\ell(\mathbf{n}+i\mathbf{a}) \leq \Delta\ell(\mathbf{m}+j\mathbf{a})$. Combined with Lemma III.1, it follows that, $\Delta\ell(\mathbf{n}+(i+k)\mathbf{a}) \leq \Delta\ell(\mathbf{m}+(j-k)\mathbf{a})$ for $0 \leq k < j$. Thus between the two cells all contexts with counts (a_0, a_1), may be mapped to one of these without increase of the adaptive code length. This may be repeated for all instances of contexts having identical occurrence counts.

Now consider contexts $\mathbf{c}$ with identical occurrence count ratios, $(n_0, n_1) = k(a_0, a_1), k>0$. These may be decomposed into counts $(n_0, n_1) = k'(a_0', a_1'), k'>0$, where a_0' and a_1' are relatively prime. Perceiving k' as a number of contexts, the case can be treated as the equal count case above. As the k' contexts may be combined for any initial combination of k values, the contexts with identical occurrence count ratios may also be mapped to the same contexts without increase in adaptive code length.

Lemma III.2 implies that there is no increase in the adaptive code length by mapping contexts having the same occurrence count ratio into the same cell. An optimal context quantizer $\bar{Q}(\mathbf{c})$ may therefore be described by a compound mapping, where in the first step all contexts with identical count ratio are mapped to the same cell, and their occurrence counts summed. This first step is referred to as a prequantizer and it may be used to reduce the complexity of the context quantization. The second step is to perform context quantization on the resulting merged counts. (Lemma III.1 and thereby also Lemma III.2 generalizes to larger alphabets. In Lemma III.1, the code lengths may be written as $l(\mathbf{n}+d\mathbf{a})$, where $\mathbf{n}$ and $\mathbf{a}$ represents occurrence counts of the symbol of the alphabet, $\mathcal{Y}$. The ratios may be represented in a space of dimensions$|\mathcal{Y}|-1$. We focus on the simpler binary case.) Encouraged by Lemma III.2, we will take this one step further and conjecture that an optimal context quantizer for conditioning binary symbols as defined above may be described by contiguous intervals on the unit interval $[0,1]$ after initially mapping all raw contexts onto the unit interval according to their value of occurrence count ratios $n_0/(n_0+n_1)$. We note that in this case the mapping onto the unit interval is independent of δ but the context quantization solution need not be.

IV. CQ Design Algorithm for Minimum Adaptive Code Length

In this section we present a graph-theoretical algorithm to design minimum adaptive code length context quantizer under the constraint of convex partition of the unit interval. This constraint will not preclude optimality if our conjecture made at the end of the preceding section is true. If not, it may be used as an approximation.

Given a binary data sequence, y^I, and the corresponding contexts, $\mathbf{c}^I$, we introduce variables z representing conditional probabilities based on occurrence counts (using $\delta = 0$ in (3)): Let $z = \hat{P}(y=0|\mathbf{c})$ denote the conditional probability of $y=0$ given the context $\mathbf{c}$, and let $z_i = \hat{P}(y=0|\mathbf{c}_i)$ be the conditional probability of a specific projected context (or context class). We sort the projected contexts such that $\hat{P}(0|\mathbf{c}_i) < \hat{P}(0|\mathbf{c}_j)$ (using $\delta=0$ in (3)) if and only if (the sorted indices) $i<j$, or $z_1 < z_2 < \cdots < z_{\mathcal{I}}$, where $\mathcal{I} < I$ is the number of distinct projection value of $\hat{P}(0|\mathbf{c}_i)$ of $\mathbf{c}^I$. The sorting does not affect the adaptive code length of a context quantizer because the code length depends only on the counts of 0's and 1's not on their occurrence order within a conditioning state.

Now we construct a directed graph $G\langle V, E\rangle$ on the projected raw contexts in the unit interval. Each distinct projected value z_i is represented by a node in the vertex set V. There is a directed edge $e_{ij} \in E$ from node z_i to node z_j, $i<j$. The strict inequality $z_i < z_j$ makes the graph $G\langle V, E\rangle$ acyclic. Each edge $e_{ij} \in E$ is assigned a weight $L(z_i, z_j]$, which is the adaptive code length of the quantizer cell $(z_i, z_j]$. For convenience we also introduce into V a dummy node $z_0 = -\epsilon$. The above construction establishes a one-to-one correspondence between an M-cell context quantizer and an M-link path from the source z_0 to the destination $z_{\mathcal{I}}$ in the acyclic graph $G\langle V, E\rangle$. It follows from the definition of the edge weights that a minimum adaptive code length context quantizer of convex codecells corresponds to a shortest path from z_0 to $z_{\mathcal{I}}$ in $G\langle V, E\rangle$. The standard shortest path algorithm can then be employed to solve the problem of minimum adaptive code length context quantization. The successive edges of this shortest path will determine the sequence of thresholds $\{q_m\}$, $1 \leq m < M$, of the optimal context quantizer. Furthermore, the shortest path algorithm will also automatically determine the optimal number M of quantizer cells for minimum adaptive code length, given y^I, $\mathbf{c}^I$, and δ.

Next we show how to compute the weight of an edge $L(q_{m-1}, q_m]$ from the count statistics of y^I and $\mathbf{c}^I$. By projecting the raw contexts onto the unit interval, the adaptive probability estimates in the quantized contiguous cells used to calculate the adaptive code length may be expressed by

$$\hat{P}_i(y|z \in (q_{m-1}, q_m]) = \frac{n_i(y,m)+\delta}{n_i(0,m)+n_i(1,m)+2\delta}, \quad (4)$$

where $\delta \geq 0$ is a parameter of the estimator and

$$n_i(y,m) = |\{t<i : y_t = y, z_t \in (q_{m-1}, q_m]\}|$$

is the number of occurrences of y in conditioning state m in the input sequence y^{i-1} prior to y_i. The adaptive code length of the quantizer cell $(q_{m-1}, q_m]$, $L(q_{m-1}, q_m]$, i.e.

$$-\sum_{i=1}^{\mathcal{I}} 1\{z_i \in (q_{m-1}, q_m]\} \log \hat{P}_i(y_i|z_i \in (q_{m-1}, q_m]).$$

can be calculated by (2). This requires the counts $n_0(m)$ and $n_1(m)$ of 0's and 1's that fall into the interval $(q_{m-1}, q_m]$. By precomputing the cumulative counts:

$$N_0(\tau) = |\{y_t = 0 | z_t \in (0,\tau]\}|, \quad N_1(\tau) = |\{y_t = 1 | z_t \in (0,\tau]\}|, \quad (5)$$

we simply have

$$n_0(m) = N_0(q_m) - N_0(q_{m-1}), \quad n_1(m) = N_1(q_m) - N_1(q_{m-1}). \tag{6}$$

With the counts $n_0(m)$ and $n_1(m)$, one can use (2) to compute the adaptive code length $L(q_{m-1}, q_m]$ in only two additions, using two tables of $O(I)$ entries each indexed by $n_0(m)$ (and $n_1(m)$) and $n_0(m) + n_1(m)$, respectively.

Summarizing the above developments, the process of designing the minimum adaptive code length quantizer for given y^I, $\mathbf{c}^I$ and δ consists of three steps: 1) Acquire contexts and counts and order the contexts by occurrence count ratios. 2) Merge contexts with identical count ratios and create tables for quantities $N_0(\tau)$, $N_1(\tau)$, $\sum_{j=0}^{n} \log(2\delta + j)$, and $\sum_{j=0}^{n} \log(\delta + j)$ as required by (2). 3) Solve the underlying shortest path problem to determine the optimal quantizer thresholds $\{q_m\}$.

Step 1 involves sorting, and hence takes $O(\mathcal{I} \log \mathcal{I})$ time. Upon the sorting of $\mathbf{c}_i$, $N_0(\tau)$ and $N_1(\tau)$, over all $0 < \tau \leq \mathcal{I}$, can be computed in $O(\mathcal{I})$ time. Assuming the tables for $\sum_{j=0}^{n} \log(2\delta + j)$ and $\sum_{j=0}^{n} \log(\delta + j)$ are given (or replaced by Stirlings approximation for large values [13]) each code length value can be calculated in constant time. Upon the completion of step 2, the weight $L(z_i, z_j)$ of any edge $e_{ij} \in E$ can be computed in $O(1)$ time by (2) and (6) in step 3. Since the acyclic directed graph $G\langle V, E\rangle$ has $O(\mathcal{I}^2)$ edges and all edge weights can be found in $O(\mathcal{I}^2)$ time as explained above, a shortest path from z_0 to $z_{\mathcal{I}}$ and thus the corresponding optimal context quantizer can be computed in $O(\mathcal{I}^2)$ time (using the standard time complexity result $O(\mathcal{I}^2)$ for the shortest path problem).

Also, by exploiting the special structure of the graph that all nodes obey a total order, one needs only the $O(\mathcal{I})$ space for cumulative counts $N_0(\tau)$ and $N_1(\tau)$ to represent the complete directed acyclic graph $G\langle V, E\rangle$, even though it has $O(\mathcal{I}^2)$ edges.

The prequantization of merging raw contexts of equal count ratio also reduces the complexity of context quantizer design. In [13] the number $\mathcal{I}$ of contexts with distinct counts was upper bounded by $O(I^{2/3})$ in the size of the data set I. The tables for $\sum_{j=0}^{n} \log(2\delta + j)$ and $\sum_{j=0}^{n} \log(\delta + j)$ can be created in $O(I)$ time. Since the shortest path algorithm has complexity $O(\mathcal{I}^2)$, the overall design complexity becomes $O(I^{4/3})$ in the length of the data set.

There is an important difference between the above proposed algorithm and that in [11]. The former considers the model cost by optimizing the number of quantizer cells M with respect to the estimator parameter δ and the message length I, whereas the latter does not although it was meant to code real binary images. The design algorithm of [11] uses minimum conditional entropy as its objective function, and it can only optimize the context quantizer for a predetermined number, M, of conditioning states, otherwise the solution could have as many states as the number of data points.

On the other hand, Greene *et al.* showed that the objective function of conditional entropy has a so-called Monge property [11]. This property allows the use of fast matrix search technique to speed up the dynamic programming algorithm for quantizer design [14]. Unfortunately, the property does not hold if the objective is to minimize the adaptive code length. A counter example is presented below.

Example IV.1: Consider three contexts with counts (1,5), (1,4) and (1,3) and the adaptive code length with $\delta = 1$. Denote these three ordered pairs of counts (a_0, a_1), (b_0, b_1) and (c_0, c_1), respectively. The Monge property implies $\ell(a_0 + b_0 + c_0, a_1 + b_1 + c_1) + \ell(b_0, b_1) \geq \ell(a_0 + b_0, a_1 + b_1) + \ell(b_0 + c_0, b_1 + c_1)$. This may also be written as $\ell(a_0 + b_0 + c_0, a_1 + b_1 + c_1) - \ell(a_0 + b_0, a_1 + b_1) \geq \ell(b_0 + c_0, b_1 + c_1) - \ell(b_0, b_1)$, and interpreted as coding (c_0, c_1) initialized by $(a_0 + b_0, a_1 + b_1)$ (left hand side) or (b_0, b_1) (right hand side). For the contexts above $\ell(a_0 + b_0 + c_0, a_1 + b_1 + c_1) - \ell(a_0 + b_0, a_1 + b_1) = \ell(3, 12) - \ell(2, 9) = 3.46$ bits and $\ell(b_0 + c_0, b_1 + c_1) - \ell(b_0, b_1) = \ell(2, 7) - \ell(1, 4) = 3.58$ bits, thus violating the Monge inequality. (The Monge inequality could also turn the other way, but clearly asymptotically it should be as stated above.)

The Monge property is a sufficient but not a necessary condition. The counter example above still does not rule out the possibility that the fast matrix search technique can be applied to minimize the adaptive code length.

This section has considered the problem of optimizing a context quantizer for a given data set.

V. Comments on application

The adaptive code length defined by (3) does not include the model cost of coding Q. Thus for actual coding of y^I, the minimum value of (3) in Q and M expresses a lower bound for the context-based adaptive code length given the set of raw contexts $\mathbf{c}^I$, under the constraint that the quantizer cells are contiguous intervals on the values of $z = \hat{P}(y|\mathbf{c})$. This lower bound can be used to evaluate heuristic context quantizers in practice, and it would be achieved if we had the foresight to choose the optimal Q and M before processing y^I.

For actual coding purposes on a single data set either the context quantizer should be included as a preamble to the code string or the CQ design could be recalculated (at given intervals) on the causal part of the data.

A probably more interesting practical approach is to design the context quantizer on a training set $y^N, \mathbf{c}^N$, and apply it to code another data set $y^I, \mathbf{c}^I$. Using adaptive probability estimates provides robustness to mismatch between the statistics of the input sequence $y^I, \mathbf{c}^I$ and the off-line estimate of the distribution $\hat{P}(y|\mathbf{c})$. Such a mismatch is likely risk because of the estimation error inherent in $\hat{P}(y|\mathbf{c})$.

The minimum adaptive code length CQ principle (and min. conditional entropy CQ) were implemented and applied to optimize contexts in arithmetic coding of MPEG4 α-plane sequences [13]. This experiment demonstrated that an optimized context quantizer can significantly improve the performance of context-based adaptive arithmetic coding.

Appendix 1

Proof of Lemma III.1. Define $\Delta\ell(s) = \ell(n_0+(s+1)a_0, n_1+(s+1)a_1) - \ell(n_0+sa_0, n_1+sa_1)$, $s \geq 0$. The proof shall show that this function is non-increasing in $s \geq 0$. Define the redundancy of $\Delta\ell(s)$ as $R(s) = \Delta\ell(s) - (a_0+a_1)H(a_0/(a_0+a_1))$, where $H(p) = -p\log p - (1-p)\log(1-p)$. Note for given (a_0, a_1) that the last term of $R(s)$ is a constant. Expressing $\ell()$ by (2) and taking the derivative with respect to s gives

$$\frac{\delta R(s)}{\delta s} = \sum_{i=|n|}^{|n|+|a|-1} \frac{a_0+a_1}{i+2\delta+s(a_0+a_1)} - \sum_{j=n_0}^{n_0+a_0-1} \frac{a_0}{j+\delta+sa_0} - \sum_{k=n_1}^{n_1+a_1-1} \frac{a_1}{k+\delta+sa_1} \quad (7)$$

where $|n| = n_0+n_1$ and $|a| = a_0+a_1$. The summands of each of the sums above may be written as $1/x_i$, $1/x_j$, and $1/x_k$, respectively. Assume that $n_0+\delta = qa_0$ and $n_1+\delta = qa_1$, $q > 0$. We shall refer to this as perfect initialization. In this case the summands may be written $x_i = i/(a_0+a_1) + s', 0 \leq i < a_0+a_1$, $x_j = j/a_0 + s', 0 \leq j < a_0$, and $x_k = k/a_1 + s', 0 \leq k < a_1$, $s' > s$. It may be shown that ordering the elements of x_j and x_k gives a set where the value of each element with order i is less than or equal to the value of element x_i. For an easier generalization, we shall compare each element of x_i, in order, with a weighted sum of elements of x_j and x_k. The elements of each set are taken in order. The elements of the x_j and x_k sets are weighted by $a_0/(a_0+a_1)$ and $a_1/(a_0+a_1)$, respectively. As s' is a constant, the term may be disregarded, introducing $x'_i = i/(a_0+a_1), 0 \leq i < a_0+a_1$, $x'_j = j/a_0, 0 \leq j < a_0$, and $x'_k = k/a_1, 0 \leq k < a_1$. For each i, one or two elements from each of the x'_j and x'_k sets are weighted, the larger element from each set having the values $\lceil (i+1)a_0/(a_0+a_1) - 1 \rceil$ and $\lceil (i+1)a_1/(a_0+a_1) - 1 \rceil$, respectively. ($\lceil x \rceil$ is the smallest integer not less than x.) Inserting these terms in the weighted sum, $w(i)$, of the x'_j and x'_k elements to be compared with x'_i provides the bound

$$\begin{aligned} w(i) &\leq \frac{a_0}{a_0+a_1}\Big\lceil \frac{(i+1)a_0}{a_0+a_1} - 1\Big\rceil \frac{1}{a_0} + \frac{a_1}{a_0+a_1}\Big\lceil \frac{(i+1)a_1}{a_0+a_1} - 1\Big\rceil \frac{1}{a_1} \\ &\leq \frac{i}{a_0+a_1} = x'_i, 0 \leq i < a_0+a_1. \end{aligned} \quad (8)$$

The convexity of $1/x$ ensures that for each $1/x_i$ in (7) there corresponds a weighted contribution of $1/x_j$ and $1/x_k$ which is not less. Thus the derivative $\frac{\delta R(s)}{\delta s}$ (7) is non-positive for all values of $s > 0$ in this special case. Further it may be noted that for $i = j = k = 0$, $x_i = x_j = x_k = s'$ in this special case.

Now consider the general case. The elements have the forms: $x_i = i/(a_0+a_1) + s_1$, $x_j = j/a_0 + s''(a_0+a_1)/a_0$ and $x_k = k/a_1 - s''(a_0+a_1)/a_1$, where s_1 and s'' are constants, and s_1 is positive. Inserting the elements gives

$$\sum_{i=|n|}^{|n|+|a|-1} x_i = \sum_{j=n_0}^{n_0+a_0-1} x_j + \sum_{k=n_1}^{n_1+a_1-1} x_k + 1/2.$$

For given n_0, n_1, a_0, a_1 and δ we consider the corresponding case of perfect initialization where the values of x_i are fixed having the same values, i.e. q of the perfect initialization is given by $q(a_0+a_1) = n_0+n_1+\delta$. Thus the x_i values will take on the same values as in the corresponding case of perfect initialization. Compared to the perfect initialization with the same values of x_i, the set of values of x_j and x_k will be moved in opposite directions, each set offset by a constant inversely proportional to the weights used in the weighted sum. Thus the offsets will cancel out in each of the weighted sums leading to the same results, prior to the application of Jensen's inequality, as in the case of perfect initialization. Again the convexity of $1/x$ ensures that the derivative remains non-positive also in the general case. This completes the proof.

REFERENCES

[1] J. Rissanen, A universal data compression system, *IEEE Trans. Information Theory*, 29(5):656–664, Sept. 1983.

[2] M.J. Weinberger, J. Rissanen and M. Feder, A universal finite memory source, *IEEE Trans. Information Theory*, 41(3):643–652, May 1995.

[3] F.M.J. Willems, Y.M. Shtarkov and T.J. Tjalkens, "The context-tree weighting method: basic properties", *IEEE Trans. Info. Theory*, vol. 41, pp. 653-664, May 1995.

[4] M.J. Weinberger, J. Rissanen and R.B. Arps, Applications of universal context modeling to lossless compression of gray-scale images, *IEEE Trans. Image Proc.*, 5(4):575–586, April 1996.

[5] JBIG, "Progressive bi-level image compression", *ISO/IEC Int. Standard 11544*, 1993.

[6] X. Wu, "Lossless compression of continuous-tone images via context selection and quantization", *IEEE Trans. on Image Proc.*, vol. 6, no. 5, pp. 656–664, 1996.

[7] JPEG-LS, "Lossless and near-losless compression of continous-tone still images", *ISO/IEC Int. Standard 14495*, 1999.

[8] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS", *IEEE Trans. Image Processing*, 9 (2000), no. 8, pp. 1309–1324.

[9] D. Taubman, "High performance scalable image compression with EBCOT", *IEEE Trans. on Image Proc.*, vol. 9, no. 7, pp. 1158–1170, 2000.

[10] X. Wu. "Context quantization with fisher discriminant for adaptive embedded wavelet image coding", *Proc. of 1999 Data Compression Conference*, pp. 102-111, Mar. 1999.

[11] D. Greene, F. Yao, T. Zhang, "A linear algorithm for optimal context clustering with application to bi-level image coding", *Proc. Int'l. Conf. Image Proc. 1998*,

[12] B. Martins and S. Forchhammer, "Tree coding of bilevel images", *IEEE Trans. Image Processing*, vol. 7, no. 4, April 1998, pp. 517-528.

[13] S. Forchhammer, X. Wu and J. D. Andersen, "Optimal context quantization in lossless compression of image data sequences", *IEEE Trans. Image Processing*, April 2004, pp. 509–517.

[14] X. Wu, "Optimal quantization by matrix-searching", *Journal of Algorithms*, vol. 12, no. 4, pp. 663-673, Dec. 1991.