CONDITIONAL WEIGHTED UNIVERSAL SOURCE CODES: SECOND ORDER STATISTICS IN UNIVERSAL CODING

Michelle Effros

Dept. of Electrical Engineering, 136-93
California Institute of Technology
Pasadena, CA 91125
effros@caltech.edu

ABSTRACT

We consider the use of second order statistics in two-stage universal source coding. (Examples of two-stage universal codes include the weighted universal vector quantization (WUVQ) [2, 3, 4], weighted universal bit allocation (WUBA) [5, 4], and weighted universal transform coding (WUTC) [6, 4] algorithms.) The second order statistics are incorporated in two-stage universal source codes in a manner analogous to the method by which second order statistics are incorporated in entropy constrained vector quantization (ECVQ) to yield conditional ECVQ (CECVQ) [1]. In this paper, we describe an optimal two-stage conditional entropy constrained universal source code along with its associated optimal design algorithm and a fast (but non-optimal) variation of the original code. The design technique and coding algorithm here presented result in a new family of conditional entropy constrained universal codes including but not limited to the conditional entropy constrained WUVQ (CWUVQ), the conditional entropy constrained WUBA (CWUBA), and the conditional entropy constrained WUTC (CWUTC). The fast variation of the conditional entropy constrained universal codes allows the designer to trade off performance gains against storage and delay costs. We demonstrate the performance of the proposed codes on a collection of medical brain scans. On the given data set, the CWUVQ achieves up to 7.5 dB performance improvement over variable-rate WUVQ and up to 12 dB performance improvement over ECVQ. On the same data set, the fast variation of the CWUVQ achieves identical performance to that achieved by the original code at all but the lowest rates (less than 0.125 bits per pixel).

1. INTRODUCTION

The universal source coding literature addresses the problem of code design for applications where the statistics of the source to be compressed are unavailable at design time. A universal code is a single code that achieves optimal performance on every source within some broad class of possible sources. Optimal performance is achieved asymptotically as the data sequence length and quantizer vector dimension grow without bound.

This material is based upon work supported by NSF Grant No. MIP-9501977

In [2, 3, 4], Chou, Effros, and Gray introduce a weighted universal vector quantizer (WUVQ) and its associated optimal design algorithm. The WUVQ uses a two-stage coding strategy to achieve universal performance. In the two-stage approach, each source description contains two sections or stages. The first stage source description specifies which code, from some given family of vector quantizers, will be used to compress the observed source. The second-stage description is the data sequence encoded using the code specified in the first-stage description.

In order to prove a code universal, one must show that as the source coding dimension grows without bound the code's performance approaches the theoretically optimal performance. Unfortunately, the computational expense of a vector quantizer grows exponentially in the vector dimension. Thus while in theory universal codes encode vectors of infinite dimension, in practice, "universal" quantizers are typically implemented with small vector dimensions.

Recent efforts in practical universal source coding algorithms have concentrated on achieving reasonable complexity codes with high effective vector dimensions. One means of achieving greater performance at reasonable computational expense is to replace the WUVQ's collection of vector quantizers with a collection of transform codes that achieve higher performance for a given computational expense. One example of a two-stage code of this type is the weighted universal bit allocation algorithm (WUBA) [5, 4]. WUBA is a two-stage JPEG-style code which replaces JPEG's single quantization matrix with an optimal collection of quantization matrices. A second example is the weighted universal transform coding algorithm (WUTC) [6, 4], which uses an optimal family of transform codes in the two-stage coding framework. (A transform code contains a transform and an associated bit allocation.) Both WUBA and WUTC provide performance improvements over the WUVQ algorithm due to the ability of each to use higher effective vector dimensions with reasonable computational expense.

In this paper, we present another means of improving two-stage coding performance while maintaining reasonable computational expense. The approach described can be applied in any two-stage coding framework. For simplicity, we here demonstrate the algorithm in the WUVQ domain. In section 2, we briefly describe the WUVQ algorithm and the optimal design algorithm for two-stage universal source codes. In section 3, we describe the conditional entropy con-

strained two-stage universal coding paradigm and its associated optimal design algorithm and fast variation. Section 4 contains experimental results.

2. THE WUVQ ALGORITHM AND TWO-STAGE UNIVERSAL SOURCE CODE DESIGN

Consider any family $\mathcal{C}^l$ of l-dimensional source codes (e.g., the family of variable-rate vector quantizers, the family of JPEG-style codes, or the family of transform codes). Each quantizer $C = \beta \circ \alpha \in \mathcal{C}^l$ contains an encoder $\alpha : \mathcal{X}^l \to \mathcal{S}$ and decoder $\beta : \mathcal{S} \to \hat{\mathcal{X}}^l$, which together map the input space $\mathcal{X}^l$ of possible data vectors to the output space $\hat{\mathcal{X}}^l$ of possible reproductions by way of a binary prefix code $\mathcal{S}$. For any $x^l \in \mathcal{X}^l$ and $C \in \mathcal{C}^l$, let $r(x^l, C) = |\alpha(x^l)|$ and $d(x^l, C) = d(x^l, \beta(\alpha(x^l)))$ represent the rate and distortion associated with encoding data vector x^l with code C.

Given an input distribution on $\mathcal{X}^l$ (or a representative training sequence), for each family of codes $\mathcal{C}^l$, techniques exist for designing a good code $C \in \mathcal{C}^l$ to match the known input statistics. For example, if $\mathcal{C}^l$ is the family of l-dimensional vector quantizers, then the generalized Lloyd algorithm provides a means for designing an optimal vector quantizer to match a given training sequence. Unfortunately, for any family of codes, optimal code design is data specific, and thus traditional compression techniques fail in applications where the source to be compressed is unknown at design time or time-varying. The two-stage approach to source coding addresses this problem.

In two-stage coding, we replace any single code $C \in \mathcal{C}^l$ with a collection of codes designed to do well across a variety of possible sources. Using the quantization interpretation of a two-stage code [3], we consider this collection to be a *codebook* of codes. Thus we define a "first-stage quantizer" $\tilde{\beta} \circ \tilde{\alpha}$, with encoder $\tilde{\alpha} : \mathcal{X}^{nl} \to \tilde{\mathcal{S}}$ and decoder $\tilde{\beta} : \tilde{\mathcal{S}} \to \mathcal{C}$ that together map the input space $\mathcal{X}^{nl}$ of data blocks to the output space $\{\tilde{\beta}(\tilde{s}) \in \mathcal{C}^l : \tilde{s} \in \tilde{\mathcal{S}}\}$ of possible source codes by way of the first-stage entropy code $\tilde{\mathcal{S}}$. The first-stage encoder chooses for each nl-block a single code. We then use the chosen code to encode each of the n l-vectors in x^{nl} in the second-stage description.

The total distortion associated with encoding data block x^{nl} with code $\tilde{\beta}(\tilde{\alpha}(x^{nl}))$ is

$$d(x^{nl}, \tilde{\beta}(\tilde{\alpha}(x^{nl}))) = \sum_{i=1}^{n} d(x_i^l, \tilde{\beta}(\tilde{\alpha}(x^{nl}))).$$

The total rate associated with encoding x^{nl} includes both the rate associated with describing a code in the collection and the rate associated with using the chosen code. Thus

$$r(x^{nl}, \tilde{\beta}(\tilde{\alpha}(x^{nl}))) = |\tilde{\alpha}(x^{nl})| + r'(x^{nl}, \tilde{\beta}(\tilde{\alpha}(x^{nl}))),$$

where $|\tilde{\alpha}(x^{nl})|$ is the rate associated with the first-stage description and $r'(x^{nl}, \tilde{\beta}(\tilde{\alpha}(x^{nl}))) = \sum_{i=1}^{n} r(x_i^l, \tilde{\beta}(\tilde{\alpha}(x^{nl})))$ is the rate associated with the second-stage description using code $\tilde{\beta}(\tilde{\alpha}(x^{nl}))$.

Then, using a Lagrangian in order to minimize the distortion subject to a constraint on the rate, the optimal first-stage encoder $\tilde{\alpha}^\star$ for a given collection of codes $\tilde{\beta}$ is

$$\tilde{\alpha}^\star = \arg\min_{\tilde{s} \in \tilde{\mathcal{S}}} [d(x^{nl}, \tilde{\beta}(\tilde{s})) + \lambda r(x^{nl}, \tilde{\beta}(\tilde{s}))]$$

for every x^{nl}. We call the optimal first-stage encoder a *nearest neighbor* encoder.

Likewise, the optimal first-stage decoder $\tilde{\beta}^\star$ for a given first-stage encoder $\tilde{\alpha}$ satisfies

$$\tilde{\beta}^\star(\tilde{s}) = \arg\min_{C \in \mathcal{C}^l} E\left[d(X^{nl}, C) + \lambda r(X^{nl}, C) \middle| \tilde{\alpha}(X^{nl}) = \tilde{s} \right]$$

for every $\tilde{s} \in \tilde{\mathcal{S}}$. We call the process of designing the optimal first-stage decoder *decoding to the centroid*. If $\mathcal{C}^l$ is the family of vector quantizers, this step may be accomplished using the generalized Lloyd algorithm [3, 4]; if $\mathcal{C}^l$ is the family of JPEG-style codes, this step may be accomplished using an optimal bit-allocation design algorithm [5, 4]; and if $\mathcal{C}^l$ is the family of transform codes, this step may be accomplished using an optimal transform code design algorithm [6, 4].

The optimal design algorithm is an iterative descent technique. We initialize the algorithm with an arbitrary prefix code $\tilde{\mathcal{S}}$ and collection $\{\tilde{\beta}(\tilde{s}) : \tilde{s} \in \tilde{\mathcal{S}}\}$ of codes in $\mathcal{C}^l$. Each iteration requires three steps, enumerated below.

1. *Nearest Neighbor Encoding*
   Optimize the first-stage encoder $\tilde{\alpha}$ for the given first-stage decoder $\tilde{\beta}$ and prefix code $\tilde{\mathcal{S}}$.
2. *Decoding to the Centroid*
   Optimize the first-stage decoder $\tilde{\beta}$ for the new first-stage encoder and the given first-stage prefix code $\tilde{\mathcal{S}}$.
3. *Optimizing the Prefix Code*
   Optimize the first-stage prefix code $\tilde{\mathcal{S}}$ for the new first-stage encoder and decoder. The optimal prefix code $\tilde{\mathcal{S}}^\star$ for a given first-stage encoder $\tilde{\alpha}$ and decoder $\tilde{\beta}$ is the entropy code matched to the probabilities $P[\tilde{\alpha}(X^{nl}) = \tilde{s}]$, for which the ideal code-lengths are

$$|\tilde{s}^\star| = -\log P[\tilde{\alpha}(X^{nl}) = \tilde{s}].$$

Each step of the algorithm decreases the expected value of the Lagrangian performance measure. Since the Lagrangian cannot be negative, the algorithm is guaranteed to converge.

3. CONDITIONAL TWO-STAGE UNIVERSAL CODES

Conditional two-stage coding addresses the desire for higher dimensional codes with reasonable computational expense by incorporating the information in the joint probability mass function of the two-stage code's codebooks. This information is incorporated in a manner analogous to that used to generalize ECVQ to CECVQ [1].

Let $\{x_1^{nl}, \ldots, x_K^{nl}\}$ be the incoming sequence of data vectors and $\mathcal{C}_1 \times \mathcal{C}_2 \times \ldots \times \mathcal{C}_K$ be a "product codebook of codebooks," with $\mathcal{C}_i$ an nl-dimensional two-stage code containing M_i codebooks. Let $P(c_1, c_2, \ldots, c_K)$ be the probability that a sequence x^{Knl} is encoded using the second-stage code

$c_i \in C_i$ to encode vector x_i^{nl}. Then the optimal entropy-constrained encoding of x^{Knl} is the encoding that minimizes the Lagrangian

$$\sum_{i=1}^{K} d(x_i^{nl}, c_i) + \lambda \left[\sum_{i=1}^{K} r'(x_i^{nl}, c_i) + \log P(c_1, c_2, \ldots, c_K) \right].$$

If $P(c_1, c_2, \ldots, c_K) = \Pi_{i=1}^{k} P(c_i)$ for all $(c_1, \ldots, c_K)$, then

$$\begin{aligned} & \sum_{i=1}^{K} d(x_i^{nl}, c_i) + \lambda \left[\sum_{i=1}^{K} r'(x_i^{nl}, c_i) - \log P(c_1, c_2, \ldots, c_K) \right] \\ = & \sum_{i=1}^{K} \left[d(x_i^{nl}, c_i) + \lambda [r'(x_i^{nl}, c_i) - \log P(c_i)] \right] \end{aligned}$$

and the algorithm described in Section 2 gives the optimal performance. However, if the distribution on the codebooks is not memoryless, then incorporation of higher order statistics will yield better coding performance. A tradeoff exists between the better performance associated with higher order codes and the higher computation and storage requirements necessary to apply such a model. We here consider a simple first-order Markov model.

Let

$$P(c_1, c_2, \ldots, c_K) = P(c_1) P(c_2|c_1) \cdots P(c_K|c_{K-1}).$$

Then

$$\begin{aligned} & \sum_{i=1}^{K} d(x_i^{nl}, c_i) + \lambda \left[\sum_{i=1}^{K} r'(x_i^{nl}, c_i) - \log P(c_1, c_2, \ldots c_K) \right] \\ = & \sum_{i=1}^{K} \left[d(x_i^{nl}, c_i) + \lambda \left[r'(x_i^{nl}, c_i) - \log P(c_i|c_{i-1}) \right] \right] \\ = & \sum_{i=1}^{K} L(x_i^{nl}, c_i|c_{i-1}), \end{aligned}$$

where we define

$$L(x_i^{nl}, c_i|c_{i-1}) = d(x_i^{nl}, c_i) + \lambda [r'(x_i^{nl}, c_i) - \log P(c_i|c_{i-1})].$$

and $P(c_1|c_0) = P(c_1)$ for all c_0. The optimal encoder in the above scenario uses dynamic programming to find the optimal sequence of codes with which to encode any sequence of vectors.

For notational simplicity, let $\mathcal{C}_i = \mathcal{C}$ for all i, where $\mathcal{C}$ is a two-stage code containing M second-stage codes $C_1, \ldots, C_M$. Then for any $m \in \{1, \ldots, M\}$ and any $k \in \{1, \ldots, K\}$, let $J_k(m)$ be the optimal Lagrangian performance on the first k nl-vectors $x_1^{nl}, x_2^{nl}, \ldots, x_k^{nl}$ using any sequence of codes $c_1, c_2, \ldots, c_k$ that satisfies $c_k = C_m$. Thus

$$J_k(m) = \min_{\{c_i \in \mathcal{C}\}_{i=1}^{k-1}} \left[\sum_{i=1}^{k-1} L(x_i^{nl}, c_i|c_{i-1}) + L(x_k^{nl}, C_m|c_{k-1}) \right]$$

and $J_0(m) = 0$ for all m. Then clearly for any $1 \le k \le K$,

$$J_k(m) = \min_{m' \in \{1, \ldots, M\}} [J_{k-1}(m') + L(x_k^{nl}, C_m|C_{m'})]$$

and the optimal encoder for the entire data sequence achieves Lagrangian performance

$$J^* = \min_{m \in \{1, \ldots, M\}} J_K(m).$$

Now let I_i be the ith codebook index in the optimal encoding of the entire data set $x_1^{nl}, \ldots, x_K^{nl}$. Then

$$I_K = \arg \min_{m \in \{1, \ldots, M\}} J_K(m),$$

and the optimal encoding can then be found by backtracking. That is,

$$\begin{aligned} I_{K-1} &= \arg \min_{1 \le m \le M} [J_{K-1}(m) + L(x_K^{nl}, C_{I_K}|C_m)], \\ I_{K-2} &= \arg \min_{1 \le m \le M} [J_{K-2}(m) + L(x_{K-1}^{nl}, C_{I_{K-1}}|C_m) \\ &\quad + L(x_{K-1}^{nl}, C_{I_K}|C_{I_{K-1}})] \\ &= \arg \min_{1 \le m \le M} [J_{K-2}(m) + L(x_{K-1}^{nl}, C_{I_{K-1}}|C_m)], \end{aligned}$$

and in general,

$$I_k = \arg \min_{1 \le m \le M} [J_k(m) + L(x_{k+1}^{nl}, C_{I_{k+1}}|C_m)],$$

where we define $L(x_k^{nl}, C_{I_k}|C_m) = 0$ for all $k > K$.

The optimal design algorithm proceeds as in Section 2 except that the new optimal encoder replaces the earlier optimal encoder and the optimal entropy code design is replaced by an optimal conditional entropy code design.

While dynamic programming is computationally efficient, the storage and delay constraints associated with dynamic programming can be prohibitive when K equals the number of nl-dimensional vectors in a large image or data sequence. We therefore employ a fast version of the conditional two-stage code in this paper, which simply modifies K to be as short as necessary to make the storage and delay requirements manageable. We discuss experimentally observed tradeoffs associated with this faster version in the following section.

4. EXPERIMENTAL RESULTS

In figure 1, we compare the performance of the CWUVQ to the performance of the WUVQ and ECVQ on a collection of medical brain scans. The constants in the above algorithms are set as: vector dimension $l = 4$, first-stage coding dimension $n = 4$, and conditioning memory $K = 4096$ in the optimal code (each image is 256 pixels by 256 pixels in size, giving 4096 nl-dimensional blocks) and $K = 4$ in our implementation of the fast version. The ECVQ codebooks contain no more than 256 codewords, while WUVQ and CWUVQ contain no more than 256 codebooks, each with at most 4 codewords. Each system is trained on 20 medical brain scans and then tested on 5 scans outside of the training set. All rates are reported in terms of entropy. The CWUVQ algorithm achieves up to 7.5 dB performance improvement over WUVQ, and up to 12 dB performance improvement over ECVQ. The fast version of the CWUVQ achieves performance almost identical to that of the optimal code at all but the lowest rates (less than 0.125 bpp).

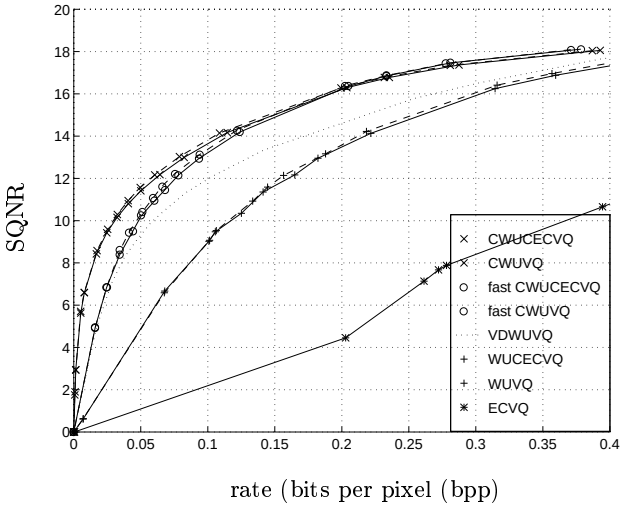

Figure 1: Comparison of SQNR results on a collection of MR brain scans. Codes with CECVQs in the second-stage (CWUCECVQ, fast CWUCECVQ, and WUCECVQ) are shown with dashed lines.

The figure also contains curves describing the performance of standard and conditional two-stage codes containing collections of CECVQs rather than collections of ECVQs. The resulting systems, labeled WUCECVQ when the first-stage does not utilize second-order statistics and CWUCECVQ when it does, yield only extremely marginal performance improvements over the analogous systems that do not use second-order statistics in the second-stage codes. This result can be explained by the small number of codewords in the second-stage codes. Also contained in the above figure is the variable dimension WUVQ (VDWUVQ), which uses a dynamic programming argument in the first-stage encoder to replace the WUVQ's fixed first-stage coding dimension n with an optimal variable first-stage coding dimension [7]. While the VDWUVQ gives significant performance improvement over WUVQ, especially at low rates, the VDWUVQ's performance is everywhere exceeded by the performance of the CWUVQ.

Figure 2 compares the image coding performance of the above CWUVQ with the image coding performance achieved using JPEG with two different sets of perceptually weighted quantization matrices and entropy codes.

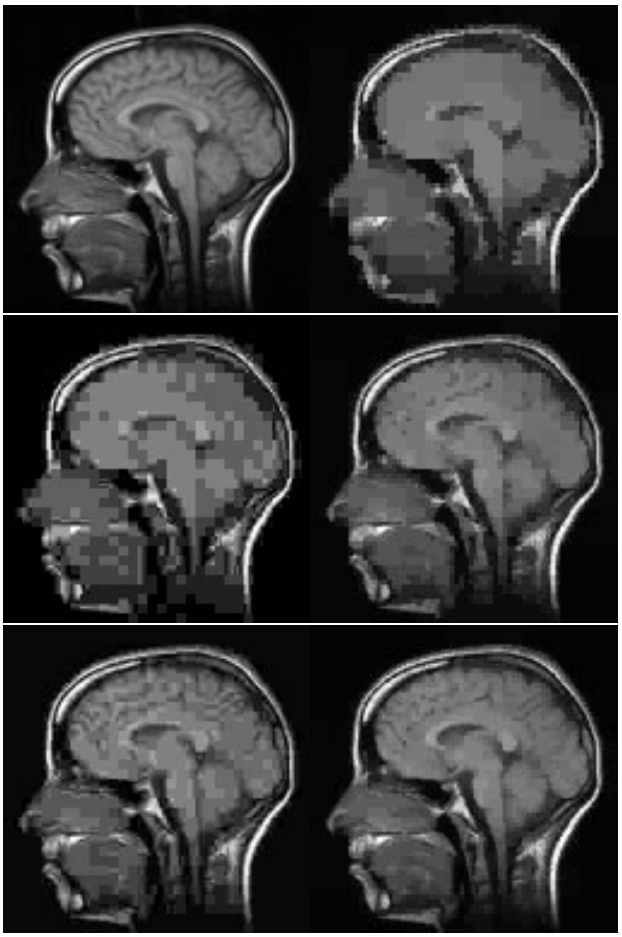

Figure 2: Compressed images. Top to bottom – Left column: Original, JPEG 0.20 bpp, JPEG 0.25 bpp. Right column: CWUVQ 0.06 bpp, CWUVQ 0.14 bpp, CWUVQ 0.24 bpp.

5. REFERENCES

[1] P. A. Chou and T. Lookabaugh. Conditional entropy-constrained vector quantization of linear predictive coefficients. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, NM, April 1990.

[2] P. A. Chou. Code clustering for weighted universal VQ and other applications. In *Proceedings of the IEEE International Symposium on Information Theory*, page 253, Budapest, Hungary, June 1991.

[3] P. A. Chou, M. Effros, and R. M. Gray. A vector quantization approach to universal noiseless coding and quantization. *IEEE Transactions on Information Theory*, IT-42(4):1109–1138, July 1996.

[4] M. Effros and P. A. Chou. Universal image compression. 1996. In preparation.

[5] M. Effros and P. A. Chou. Weighted universal bit allocation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2343–2346, Detroit,MI, May 1995. IEEE.

[6] M. Effros and P. A. Chou. Weighted universal transform coding: universal image compression with the Karhunen-Loeve transform. In *Proceedings of the IEEE International Conference on Image Processing*, Washington, D.C., October 1995. IEEE.

[7] M. Effros, P. A. Chou, and R. M. Gray. Variable dimension weighted universal vector quantization and noiseless coding. In *Proceedings of the Data Compression Conference*, pages 2–11, Snowbird, UT, March 1994. IEEE.