Network Vector Quantization *

Michael Fleming Michelle Effros

Abstract

A network source code is an optimal source code for a network. To design network source codes, we require each node to have a single encoder, which jointly encodes all messages transmitted by that node, and a single decoder, which jointly decodes all messages arriving at that node. Given a distribution over the sources, the design of the network source code jointly optimizes all encoders and decoders to obtain the best performance with respect to a user-defined priority schedule over the rates and distortions of the system.

In this paper we focus on fixed-rate codes and address the implementation of an existing design algorithm for optimal network vector quantizers. Implementing the design algorithm is not straightforward since each encoder must choose its reproduction based on the expected behavior of sources that are unknown to it. We describe a new implementation approach and demonstrate its performance on a three-node network. In addition, we extend the design algorithm to allow the decoder at each node to use side information (specifically, the messages that are to be encoded by the encoder at the same node).

1 Introduction

Any network, such as a cellular network, a computer network, or a remote sensing network, requires a good data compression system to be efficient. Using source and channel codes, the amount of data to be transmitted between the nodes can be reduced and reliability increased. Whilst there has been significant research into channel coding for networks, source coding for networks is still a largely unexplored area. In most networks, source coding is still done using a separate, independent source code for each transmitter-receiver pair. Such an approach makes design simple, but the efficiency of the network as a whole suffers. Although independent source coding removes the redundancy in each single source description, it fails to remove the redundancy resulting from correlation between sources.

In networks where source correlation is high, significant gains can be made by exploiting it. A good example is that of a sensor network in which measurements

*M. Fleming and M. Effros are with the Department of Electrical Engineering, MC 136-93, California Institute of Technology, Pasadena, CA 91125. This material is based upon work supported by the Pickering Fellowship, the F.W.W. Rhodes Memorial Scholarship, a Redshaw Award, NSF Grant No. CCR-9909026, a grant from the Lee Center for advanced networking at Caltech, and the Intel Technology for Education 2000 program.

1068-0314/01 $10.00 © 2001 IEEE

are made by spatially varying nodes in the same environment. We expect correlation between the measurements at different nodes, and we would like our source code to use this correlation to reduce the amount of data transfer between nodes.

To take full advantage of source correlation, we allow the source code to jointly encode and jointly decode messages whenever possible. Further, we optimize the code globally, over the entire network, rather than locally, over each encoder-decoder pair. The resulting code is called a *network source code.* This paper relies on the network source coding framework of [1], which can be applied to any network. Special cases include multi-resolution, multiple description, broadcast, and multiple access systems.

Since a network source code is intended to both jointly encode and jointly decode messages whenever possible, its structure differs from that of a collection of independent transmitter-receiver codes. In the independent case, each node has a separate encoder for each message it transmits, and a separate decoder for each message it receives. In contrast, a network source code has a single encoder and a single decoder at each node – the encoder jointly encodes all messages transmitted by that node, and the decoder jointly decodes all messages received by that node.

In [1], a metric for network source code optimality is given, from which the design equations for optimal fixed- and variable-rate network vector quantizers (NVQs) are derived. In this paper we focus on fixed-rate NVQs and extend NVQ design to allow each node to use the source that it encodes as side information for that node's decoder. (Since the encoder and decoder are at the same node, we assume that they can share information; this improves decoding performance if the sources encoded and decoded at that node are correlated.) More importantly, we address the issue of practical NVQ implementation, which was not treated in [1]. Issues that arise as a result of joint decoding are illustrated using a system that combines multiple access (also known as multiterminal) and Wyner-Ziv coding. New optimal VQ design algorithms for both of these systems can be extracted as special cases of the general design algorithm. Finally, we give experimental results, showing the performance gain attained by NVQ over independently designed VQs.

2 Network Description

Consider an M-node network. In the most general case, every node communicates with every other node, and a message may be intended for any subset of nodes. We denote by $X_{t,S}$ the message sent from node t to the nodes in set $S \subset \mathcal{M} = \{1, \ldots, M\}$. For example, $X_{1,\{2,4\}}$ is the message sent by node 1 to nodes 2 and 4. If S contains only one index, we write $X_{t,\{r\}} = X_{t,r}$ for simplicity. For any node $r \in S$, we denote by $\hat{X}_{t,S,r}$ the reproduction at node r of message $X_{t,S}$. Thus $\hat{X}_{1,\{2,4\},2}$ is node 2's reproduction of source $X_{1,\{2,4\}}$. This will generally differ from node 4's reproduction $\hat{X}_{1,\{2,4\},4}$ of $X_{1,\{2,4\}}$, since it is to be jointly decoded with different messages.

For each node $t \in \mathcal{M}$, let $\mathcal{S}(t)$ denote a collection of sets such that for each set $S \in \mathcal{S}(t)$, there exists a message to be described by node t to precisely the members of set $S \subseteq \mathcal{M}$. Let $\mathcal{S} = \{(t, S) : t \in \mathcal{M}, S \in \mathcal{S}(t)\}$ denote the set of indices for all messages in a given network. The vector of messages transmitted by node t is denoted

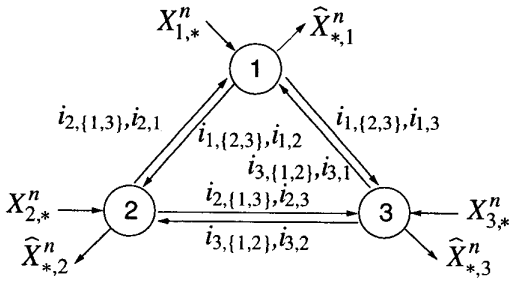

Figure 1: A general three-node network.

by $X_{t,*} = (X_{t,S})_{S \in \mathcal{S}(t)}$. Now for each $r \in \mathcal{M}$ define $\mathcal{T}(r) = \{(t,S) \in \mathcal{S} : r \in S\}$ to describe the set of messages received by node r. The vector of reconstructions at node r is denoted by $\hat{X}_{*,r} = (\hat{X}_{t,S,r})_{(t,S) \in \mathcal{T}(r)}$. Finally, let $\mathcal{T} = \{(t,S,r) : r \in \mathcal{M}, (t,S) \in \mathcal{T}(r)\}$ denote the set of all transmitter-message-receiver triples.

Figure 1 shows a general three-node network. In this network, each node transmits three messages and receives four. Node 2, for example, encodes $X_{2,*} = (X_{2,\{1,3\}}, X_{2,1}, X_{2,3})$ and decodes $\hat{X}_{*,2} = (\hat{X}_{1,\{2,3\},2}, \hat{X}_{1,2,2}, \hat{X}_{3,\{1,2\},2}, \hat{X}_{3,2,2})$. Messages $X_{2,*}$ may also be viewed as side information of potential use in decoding $\hat{X}_{*,2}$.

In this paper we consider block source codes. For each $(t,S) \in \mathcal{S}$, the source $\{X_{t,S}(i)\}_{i=1}^{\infty}$ is blocked into n-vectors $\{X^n_{t,S}(j)\}_{j=1}^{\infty}$. For any vector $X^n_{t,*}$ of source n-vectors, the encoder at node t, denoted $\alpha^n_t : \mathcal{X}^n_{t,*} \to \mathcal{C}(t,*)$, maps $X^n_{t,*}$ to a fixed-length channel codeword $c_{t,*} \in \mathcal{C}(t,*)$. Here $c_{t,*} = (c_{t,S})_{S \in \mathcal{S}(t)}$, and for each $S \in \mathcal{S}(t)$, $c_{t,S} \in \mathcal{C}(t,S)$. The codeword $c_{t,*}$ is then sent through the channel, which is assumed to noiselessly convey transmitted codeword $c_{t,S}$ to precisely the receivers $r \in S$. Thus for any $r \in \mathcal{M}$, the decoder at node r noiselessly receives the transmitted codewords $c_{*,r}$. We denote the decoder at node r by $\beta^n_r : \mathcal{C}(*,r) \times \mathcal{X}^n_{r,*} \to \hat{\mathcal{X}}^n_{*,r}$. It maps the collection of codewords $c_{*,r}$ along with side information $X^n_{r,*}$ (discussed below) to a collection of reproduction vectors $\hat{X}^n_{*,r}$ such that $\hat{X}^n_{t,S,r} \in \hat{\mathcal{X}}^n_{t,S,r}$ for each $(t,S) \in \mathcal{T}(r)$. Let $\beta^n_{t,S,r}(c_{*,r}, x^n_{r,*})$ denote the reproduction of $X^n_{t,S}$ achieved at receiver r. Then $\beta^n_r(c_{*,r}, x^n_{r,*}) = \hat{X}^n_{*,r}$ implies that $\beta^n_{t,S,r}(c_{*,r}, x^n_{r,*}) = \hat{X}^n_{t,S,r}$ for each $(t,S) \in \mathcal{T}(r)$. Note that $\beta^n_{t,S,r}$ depends on $c_{*,r}$ rather than simply $c_{t,S}$ since $c_{*,r}$ is jointly decoded.

Encoder α^n_t and decoder β^n_t sit at the same node of the network. Thus message $X^n_{t,*}$, encoded by α^n_t, is likely to be available as side information to decoder β^n_t. We allow for this possibility by including $X^n_{t,*}$ as an input to β^n_t.

We index the codewords in codebook $\mathcal{C}(t,S)$ using index set $\mathcal{I}(t,S)$. The collection $c_{t,*}$ of codewords transmitted by node t can thus be described in terms of its corresponding collection of indices $i_{t,*} \in \mathcal{I}(t,*)$, so that for each $S \in \mathcal{S}(t)$, $i_{t,S}$ is the index of $c_{t,S}$ in codebook $\mathcal{C}(t,S)$. Similarly, the collection $c_{*,r}$ of codewords received at node r can be described in terms of the corresponding indices $i_{*,r} \in \mathcal{I}(*,r)$.

For any fixed blocklength n and each $t \in \mathcal{M}$, the encoder $\alpha^n_t : \mathcal{X}^n_{t,*} \to \mathcal{C}(t,*)$ decomposes into lossy and lossless components as $\alpha^n_t = \gamma_t \circ \alpha_t$. Here $\alpha_t : \mathcal{X}^n_{t,*} \to \mathcal{I}(t,*)$ and $\gamma_t : \mathcal{I}(t,*) \to \mathcal{C}_{t,*}$, where $\circ$ denotes composition. The component α_t maps source vector $X^n_{t,*}$ to a vector of indices. This procedure is many-to-one and information

lossy in general. The component γ_t maps a vector of indices to a concatenated string of binary descriptions. This mapping is one-to-one and information lossless.

Similarly, for each $r \in \mathcal{M}$, decompose the decoder $\beta_r^n : \mathcal{C}(*,r) \times \mathcal{X}_{r,*}^n \to \hat{\mathcal{X}}_{*,r}^n$ as $\beta_r^n = \beta_r \circ \gamma_r^{-1}$, where $\gamma_r^{-1} : \mathcal{C}(*,r) \to \mathcal{I}(*,r)$ and $\beta_r : \mathcal{I}(*,r) \times \mathcal{X}_{r,*}^n \to \hat{\mathcal{X}}_{*,r}^n$. The mapping γ_r^{-1} maps each channel codeword back to its index. The mapping β_r outputs a vector of reproductions given the vector of decoded indices and the side information.

The performance of an n-dimensional fixed-rate network source code Q^n is its distortion. For each $(t,S,r) \in \mathcal{T}$, let $d_{t,S,r}^n : \mathcal{X}_{t,S}^n \times \hat{\mathcal{X}}_{t,S,r}^n \to [0,\infty)$ be a non-negative, additive distortion measure between the source and reproduction alphabets. Let P denote the distribution on sources $X_{*,*}^n = (X_{1,*}^n, \ldots, X_{M,*}^n)$ and E the expectation w.r.t. P. The expected distortion in describing n symbols from P with code Q^n is

$$(D_{t,s,r}^n(P,Q^n))_{(t,s,r)\in\mathcal{T}} = \left(Ed_{t,S,r}^n\left(X_{t,S}^n, \beta_{t,S,r}((\alpha_{t',S'}(X_{t',*}^n))_{(t',S')\in\mathcal{T}(r)}, X_{r,*}^n)\right)\right)_{(t,S,r)\in\mathcal{T}}.$$

3 NVQ Design

From [1] we have the following Lagrangian performance metric for fixed-rate codes:

$$j^n(P,\mu,Q^n) = \frac{1}{n}\sum_{(t,S)\in\mathcal{S}}\sum_{r\in S}\mu_{t,S,r}D_{t,S,r}^n(P,Q^n). \tag{1}$$

By varying the Lagrangian constants $\mu = (\mu_{t,S,r})_{(t,S,r)\in\mathcal{T}}$, we vary the relative importance of the distortions in the optimization process and design codes at different points on the convex hull of the region of achievable distortions for a given rate.

The system is optimized using the generalized Lloyd algorithm, optimizing each encoder and decoder of the system in turn while the remaining components are held fixed. We state below the necessary conditions for optimality of $\{\alpha_t\}$ and $\{\beta_r\}$ when the other system components are held fixed. Note that for fixed-rate coding, $\{\gamma_{t,*}\}$ and $\{\gamma_{*,r}^{-1}\}$ are fixed throughout the optimization process.

First, choose some $T \in \mathcal{M}$. We consider the optimal design of α_T given that $\{\alpha_t\}_{t\in\mathcal{M},t\neq T}$ and $\{\beta_r\}_{r\in\mathcal{M}}$ are fixed. The optimal encoder α_T^*, satisfies

$$\alpha_T^*(x_{T,*}^n) = \arg\min_{i(T,*)} \frac{1}{n}\sum_{r\in S':S'\in\mathcal{S}(T)}\sum_{(t,S)\in\mathcal{T}(r)} \mu_{t,S,r}E\left[d_{t,S,r}^n(X_{t,S}^n, \beta_{t,S,r}(I(*,r), X_{r,*}^n))\middle| X_{T,*}^n = x_{T,*}^n, I(T,*) = i(T,*)\right].$$

The expression inside the minimization describes the expected implications of the choice of $i(T,*)$ on the distortion achieved in reproducing $X_{t,*}^n$ for each t. The choice of $\alpha_T(x_{T,*}^n)$ affects the reproductions for $X_{t,*}^n$ such that $t \neq T$ as well as the reproductions for $X_{T,*}^n$ because each decoder β_r *jointly* maps the indices of $i(*,r) = (i(t,S))_{(t,S)\in\mathcal{T}(r)}$ to its corresponding vector of reproductions. Thus $i(T,*)$ affects *all* reproductions at any node r for which $r \in S'$ for some $S' \in \mathcal{S}(T)$. Here $I(*,r)$ is the vector of indices received at node r. The use of capitalization denotes the fact that for any $t \neq T$, $i(t,S)$ is unknown to α_T and must be treated as a random variable. The expectation is taken over the distribution imposed by $x_{T,*}^n$ on $X_{t,S}^n$, $I(*,r)$, and the

side information $X^n_{r,*}$. For any $t \neq T$, the distribution on $I(t,*)$ is governed by the corresponding (fixed) encoder $\{\alpha_t\}$ together with the conditional distributions on the inputs to that encoder, conditioned on the knowledge of the vector $x^n_{T,*}$ to be encoded.

Now fix all encoders $\{\alpha_t\}_{t\in\mathcal{M}}$ and all but one decoder $\{\beta_r\}_{r\in\mathcal{M},r\neq R}$, and consider the optimization of decoder β_R. The optimal decoder is $\beta^*_R = (\beta^*_{t,S,R})_{(t,S)\in\mathcal{T}(R)}$, where for any index vector $i(*,R) = (i(t,S))_{(t,S)\in\mathcal{T}(R)}$ and side information $x^n_{R,*}$,

$$\beta^*_{t,S,R}(i(*,R), x^n_{R,*}) = \arg \min_{\hat{x}^n \in \hat{\mathcal{X}}^n_{t,S,R}} E\left[d^n_{t,S,R}(X^n_{t,S}, \hat{x}^n) \middle| X^n_{R,*} = x^n_{R,*}, \alpha_{t',S'}(X_{t',*}) = i(t',S') \ \ \forall (t',S') \in \mathcal{T}(R)\right].$$

4 Implementing the Design Equations

In this section we discuss the implementation of the design algorithm, which was not treated in [1]. For simplicity, we focus on a three-node network, as shown in Figure 1.

The implementation of the optimal decoders follows directly from the design equations; the expected values can be evaluated directly using the training data. The implementation of the optimal encoders, however, is not so simple, because the design equation for any particular encoder involves an expectation over conditional distributions on the outputs of the other encoders, conditioned on the sample to be encoded. We must estimate these conditional distributions using the training data, and use the estimates to evaluate the expected distortions involved in the design equation.

Without loss of generality, we illustrate the design of the optimal encoders using the specific case of α^*_1. From the design equations, α^*_1 is given by:

$$\alpha^*_1(x_{1,*}) = \arg \min_{i(1,*)} \frac{1}{n} \left(\sum_{(t,S)\in\mathcal{T}(2)} \mu_{t,S,2} E\left[d_{t,S,2}(X^n_{t,S}, \beta_{t,S,2}(I(*,2), X^n_{2,*})) \middle| X^n_{1,*} = x^n_{1,*}, I(1,*) = i(1,*)\right] + \sum_{(t,S)\in\mathcal{T}(3)} \mu_{t,S,3} E\left[d_{t,S,3}(X^n_{t,S}, \beta_{t,S,3}(I(*,3), X^n_{3,*})) \middle| X^n_{1,*} = x^n_{1,*}, I(1,*) = i(1,*)\right] \right) \quad (2)$$

where $\mathcal{T}(2) = \{(1,\{2,3\}),(1,2),(3,\{1,2\}),(3,2)\}$, and $\mathcal{T}(3) = \{(1,\{2,3\}),(1,3),(2,\{1,3\}),(2,3)\}$. The first sum in (2) involves distortions for the reproductions at node 2, and the second involves distortions for the reproductions at node 3. The sums are identical in form. We consider each sum separately, breaking the network into two simpler subsystems, one corresponding to each sum, as shown in Figure 2.

One subsystem corresponds to the sum of the distortions of the reproductions made by node 2. These distortions depend on the encoding of the sources to be reproduced at node 2 and on the decoder β_2. The sources being reproduced are $X^n_{1,\{2,3\}}$ and $X^n_{1,2}$, which are encoded by α_1 into indices $(i_{1,\{2,3\}}, i_{1,2})$, and $X^n_{3,\{1,2\}}$ and $X^n_{3,2}$, which are encoded by α_3 into indices $(i_{3,\{1,2\}}, i_{3,2})$. Decoder β_2 jointly decodes these indices using $X^n_{2,*}$ as side information.

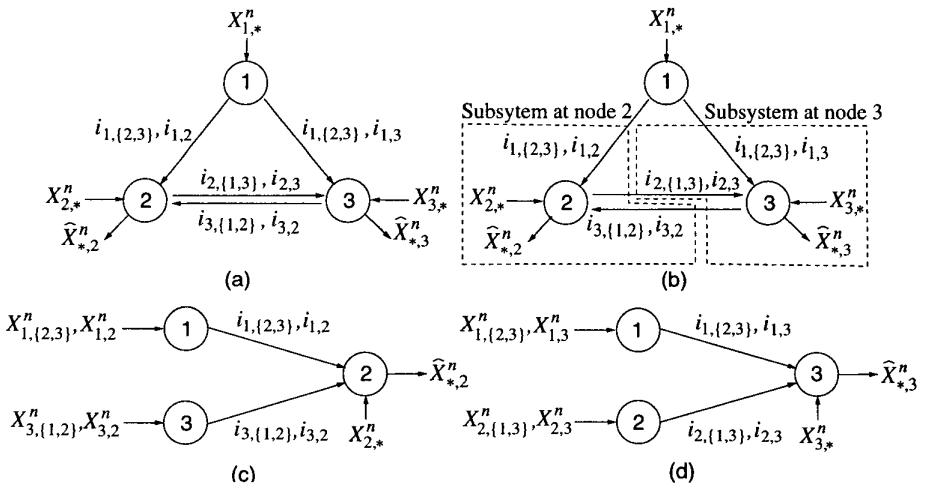

Figure 2: Optimal encoding for node 1. (a) The messages giving rise to the reproductions involved in the equation for α_1^*. (b) The estimated performance for a given index set $i_{1,*}$ can be found by summing the performance in two simpler subsystems. (c),(d) The two separate subsystems.

The subsystem for the sum over the distortions at node 3 has the same structure. The sources reproduced at node 3 are $X^n_{1,\{2,3\}}$ and $X^n_{1,3}$, which are encoded by α_1 into indices $(i_{1,\{2,3\}}, i_{1,3})$, and $X^n_{2,\{1,3\}}$ and $X^n_{2,3}$, which are encoded by α_2 into indices $(i_{2,\{1,3\}}, i_{2,3})$. Decoder β_3 jointly decodes these indices using $X^n_{3,*}$ as side information.

The two subsystems are linked by the common choice of $i_{1,*}$ (in particular, $i_{1,\{2,3\}}$) made by the encoder that we are trying to optimize. The design equation (2) tells us that the optimal choice of $i_{1,*}$ for a given input to α_1 is that which minimizes the sum of the weighted distortions over the two subsystems. In the remainder of this section, we develop an approach to estimate the appropriate sum for each subsystem as a function of $i_{1,*}$. The optimal encoder is then found by choosing the index set $i_{1,*}$ with the lowest sum distortion over the two subsystems in accordance with (2).

It can be shown that the first and second sums in (2) are the expressions that must be minimized to find the optimal encoder for the subsystems at nodes 2 and 3 respectively. We present a way to evaluate these sums by showing how to find the optimal encoder for each subsystem. Once this is done, a complete strategy for finding α_1^* has been developed. In addition, by solving the problems of how to initialize the encoders and decoder, and how to use side information in the subsystems, we solve the corresponding problems in the network system as a whole.

The subsystems can be treated simultaneously, since they have the same generic form, shown in Figure 3. For example, comparing Figure 2(c) to Figure 3, we see that when applying the discussion of the generic form to the subsystem at node 2 we should make the substitutions $(X^n_{1,\{2,3\}}, X^n_{1,2}) \leftrightarrow X$, $(X^n_{3,\{1,2\}}, X^n_{3,2}) \leftrightarrow Y$, $X_{2,*} \leftrightarrow Z$, $(i_{1,\{2,3\}}, i_{1,2}) \leftrightarrow j$, $(i_{3,\{1,2\}}, i_{3,2}) \leftrightarrow k$.

The generic subsystem combines multiple access source coding with the Wyner-

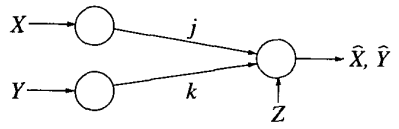

Figure 3: The subsystem for performance evaluation.

Ziv problem of decoding using side information available only at the decoder. Two independent encoders operate on sources X and Y respectively to produce indices j and k. The decoder jointly decodes these indices using side information Z which is not available to either encoder. We assume sources X, Y, and Z are correlated and are already blocked, so that samples x, y, and z are n-dimensional vectors.

The system has two encoders and one joint decoder. Encoder $\alpha_X : \mathcal{X} \to \mathcal{J}$ maps input $x \in \mathcal{X}$ to a single index $j \in \mathcal{J}$. Encoder $\alpha_Y : \mathcal{Y} \to \mathcal{K}$ maps input $y \in \mathcal{Y}$ to a single index $k \in \mathcal{K}$. The decoder β receives indices j and k and side information $z \in \mathcal{Z}$, and makes reproductions $(\hat{X}, \hat{Y})$ of sources X and Y. We impose a constraint that there should be some finite number of different reproductions $(\hat{x}, \hat{y})$ corresponding to a particular (j, k) pair. These are indexed by (j, k, ℓ), where $\ell \in \mathcal{L} = \{1, \ldots, L\}$. Each index triple uniquely specifies a decoder codeword. The decoder codewords are denoted $\beta(j, k, \ell) = (\hat{x}_{jk\ell}, \hat{y}_{jk\ell})$, with $\beta_X(j, k, \ell) = \hat{x}_{jk\ell} \in \mathcal{X}$ and $\beta_Y(j, k, \ell) = \hat{y}_{jk\ell} \in \mathcal{Y}$ specifying the reproductions individually. A separate set of codewords $\{\phi_Z(j, k, \ell)\}$ defined on alphabet $\mathcal{Z}$ is also used, not for reproduction, but for comparison to the side information to choose the best index ℓ when decoding a given (j, k) pair. We denote by $\delta : \mathcal{J} \times \mathcal{K} \times \mathcal{Z} \to \mathcal{L}$ the deterministic function that describes the decoder's choice of ℓ given received indices j and k and side information z.

We use the generalized Lloyd algorithm to design the optimal encoders and decoder of the system. Since this approach is iterative, the encoders and decoder require initialization. This can be performed by first training VQ codebooks for each source separately. We then initialize α_X and α_Y to be the nearest neighbor encoders corresponding to these two separately designed codebooks. If no side information were present, we could construct the joint decoder simply by taking the cross product of the two codebooks, but with side information we need to specify L initial codewords for each (j, k) pair. Denote by Γ the list of training set vectors for the subsystem optimization. For each (j, k) pair, we define the subset of training vectors

$$\Gamma^{j,k} = \{(x, y, z) : \alpha_X(x) = j, \alpha_Y(y) = k\}.$$

In practice, the decoder will use function $\delta(j, k, z)$ to choose between the L reproductions for a given (j, k) pair based on side information Z. Thus in training, we use only Z to choose among the L reproductions for (j, k). We implement $\delta(j, k, z)$ as a VQ encoder on codebook $\phi^{j,k} = \{(\phi(j, k, \ell)\}_{\ell=1}^{L}$ designed using the generalized Lloyd algorithm with distortion d_Z and training set $\{z : (x, y, z) \in \Gamma^{j,k}\}$. We then define the decoder β by setting $\beta(j, k, l)$ to be the centroid (w.r.t. chosen distortion measures d_X, d_Y) of the set $\{(x, y) : (x, y, z) \in \Gamma^{j,k}, \delta(j, k, z) = \ell\}$.

The performance (1) of a code Q for this system can be written as

$$j(p, \mu, Q) = \int_x \int_y \int_z p(x, y, z) \left[\mu_X d_X(x, \beta_X(\alpha_X(x), \alpha_Y(y), \delta(\alpha_X(x), \alpha_Y(y), z)))\right.$$

$$+\mu_Y d_Y(y, \beta_Y(\alpha_X(x), \alpha_Y(y), \delta(\alpha_X(x), \alpha_Y(y), z)))] \, dxdydz, \qquad (3)$$

where $p = p(x, y, z)$ is the joint distribution of (x, y, z), $\mu = (\mu_X, \mu_Y)$ is the set of Lagrangian weights used to control the importance of the terms in the optimization, and d_X and d_Y are non-negative, additive distortion measures.

It can be shown that the optimal encoder α_X^* is given by

$$\begin{aligned} \alpha_X^*(x) &= \arg\min_j \sum_k \sum_\ell \Big[\mu_X d_X(x, \beta_X(j,k,\ell)) \Pr(\alpha_Y(Y) = k, \delta(j,k,Z) = \ell | x) \\ &\quad + \mu_Y \int_{y:\alpha_Y(y)=k} d_Y(y, \beta_Y(j,k,\ell)) \Pr(Y = y, \delta(j,k,Z) = \ell | x) dy \Big]. \end{aligned} \qquad (4)$$

The expression for α_Y^* is obtained by swapping the roles of X and Y in (4).

In practice, we must estimate the expression defining the optimal encoder. The conditional probabilities $\Pr(\alpha_Y(Y) = k, \delta(j,k,Z) = \ell | x)$ and $\Pr(Y = y, \delta(j,k,Z) = \ell | x)$ must be estimated for each x using the training data. Also, the integral over y must be estimated numerically because an analytical integrand is unavailable.

We perform the estimation using a histogram technique and evaluate the integral using the known mappings for all of the training vectors. This reduces our need for knowledge of distributions on $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ down to knowledge of distributions on the much smaller space $\mathcal{X} \times \mathcal{K} \times \mathcal{L}$. Since the space $\mathcal{X}$ is, even by itself, usually very large, we approximate the quantities of interest by piecewise constant functions on $\mathcal{X}$. We partition $\mathcal{X}$ into cells $\{V_a\}_{a \in \mathcal{A}}$, $\mathcal{A} = \{1, \ldots, |\mathcal{A}|\}$, and estimate conditional distributions over $\mathcal{K} \times \mathcal{L}$ for each cell. Denote by $\phi_X : \mathcal{X} \to \mathcal{A}$ the function that maps a sample x to the appropriate partition V_a. We estimate $\Pr(\alpha_Y(Y) = k, \delta(j,k,Z) = \ell | x)$ in the first term of (4) by replacing the conditioning on x with conditioning on $\phi_X(x)$ to give an estimate $P_Y(\phi_X(x), j, k, \ell)$:

$$\begin{aligned} P_Y(\phi_X(x), j, k, \ell) &\triangleq \Pr(\alpha_Y(Y) = k, \delta(j,k,Z) = \ell | \phi_X(x)) \\ &= \frac{|\{(x', y', z') \in \Gamma : \phi_X(x') = \phi_X(x), \alpha_Y(y') = k, \delta(j,k,z') = \ell\}|}{|\{(x', y', z') \in \Gamma : \phi_X(x') = \phi_X(x)\}|} \end{aligned}$$

$P_Y(\phi_X(x), j, k, \ell)$ can be evaluated from training data, using the current (fixed) α_Y and $\delta(j,k,z)$. Similarly, we estimate the integral in the second term of (4) by

$$D_Y(\phi_X(x), j, k, \ell) \triangleq \frac{\sum_{(x', y', z') \in \Gamma : \phi_X(x') = \phi_X(x), \alpha_Y(y') = k, \delta(j,k,z') = \ell} d_Y(y', \beta_Y(j,k,\ell))}{|\{(x', y', z') \in \Gamma : \phi_X(x') = \phi_X(x)\}|}.$$

Equation (4) is then estimated as

$$\alpha_X^*(x) \simeq \arg\min_j \sum_{k,\ell} [\mu_X d_X(x, \beta_X(j,k,\ell)) P_Y(\phi_X(x), j, k, \ell) + \mu_Y D_Y(\phi_X(x), j, k, \ell)].$$

Now we consider the optimization of the decoder. To optimize the decoder when the two encoders and the functions δ are held fixed, it can be shown that each codeword $\beta(j,k,\ell)$ and $\phi(j,k,\ell)$ should be set to the centroid, with respect to the appropriate distortion measures and components (for β, components X and Y, for ϕ, component Z), of all training vectors mapped to index set (j,k,ℓ). To optimize $\delta(j,k,z)$ when all other components are held fixed we set $\delta(j,k,z)$ to be the nearest-neighbor mapping (w.r.t. d_Z) on the codewords $\phi(j,k,\ell)$, $\ell = 1, \ldots, L$.

Convergence

In the algorithm presented above, estimations were made in implementing the optimal encoder to reduce the number of conditional distributions and expected distortions required to a computationally feasible number. These estimations represent a deviation from the truly optimal encoder as specified by the generalized Lloyd algorithm, and, as a result, convergence of the algorithm is no longer guaranteed (although it is almost always observed in practice given a data set of appropriate size). We now show that by altering our performance measure, convergence can be guaranteed and computability retained at the cost of some performance gain.

Let A and B be the random variables defined by $A = \phi(X)$ and $B = \phi(Y)$. It can be shown that if $X \leftrightarrow \phi_X(x) \leftrightarrow \phi_Y(y) \rightarrow Y$ and $X, Y \leftrightarrow \phi_X(x), \phi_Y(y) \leftrightarrow Z$ form Markov chains, then $p(x, y, z, a, b) = p(x)p(a|x)p(b|a)p(y|b)p(z|a, b)$ and we could implement the optimal encoder and decoder exactly using conditional distributions that are conditioned only on $a = \phi_X(x)$ and $b = \phi_Y(y)$, as desired. These Markov properties do not hold in general, but we can make use of the results. For any distribution $p(x, y, z, a, b)$ (not necessarily satisfying the Markov properties), define $\hat{p}(p, x, y, z, a, b) = p(x)p(a|x)p(b|a)p(y|b)p(z|a, b)$ It can be shown that for any p, the corresponding $\hat{p}$ is a distribution satisfying the Markov properties. We now define a new performance measure $\hat{j}(p, \mu, Q)$ which is identical to $j(p, \mu, Q)$ in (3), but with $p(x, y, z) = p(x, y, z, \phi_X(x), \phi_Y(y))$ replaced by $\hat{p}(x, y, z, \phi_X(x), \phi_Y(y))$. The new measure $\hat{j}$ represents the expected system performance with respect to $\hat{p}$ rather than p, where $\hat{p}$ has the properties we desire. Both the optimal encoder and decoder for $\hat{j}$ can be implemented exactly (in a computationally feasible manner), and hence convergence is guaranteed. However, the system is now optimized with respect to $\hat{p}$ rather than the true joint distribution p, and does not perform as well in practice, as shown by the experimental results.

5 Experimental Results

The design algorithm is implemented for the general three-node network, and two experiments are conducted: the first examines the efficiency of network source codes as a function of source correlation, and the second compares the tradeoff in performance at each node as a function of the Lagrangian weights controlling the optimization. All experiments are conducted using fixed-rate codes at vector dimension 4, allocating 2 bits to describe each index $i_{t,S}$. Unless specified otherwise, the side information is used to choose between four possible reconstructions for each index set.

The performance gain of network source codes over independent coding is a function of source correlation. Figure 4(a) shows a plot of performance (1) in dB as a function of correlation for Gaussian sources. The training and test sets were generated such that for each sample $x_{*,*} = (x_{1,*}, x_{2,*}, \ldots, x_{M,*})$, the correlation between any two individual sources has the same value. The four samples $x_{*,*}(1), \ldots, x_{*,*}(4)$ within a blocked sample $x^4_{*,*}$ are generated independently. As the correlation between sources increases, we see that the performance of network codes improves significantly over independent codes. We see also that the use of side information (introduced in this

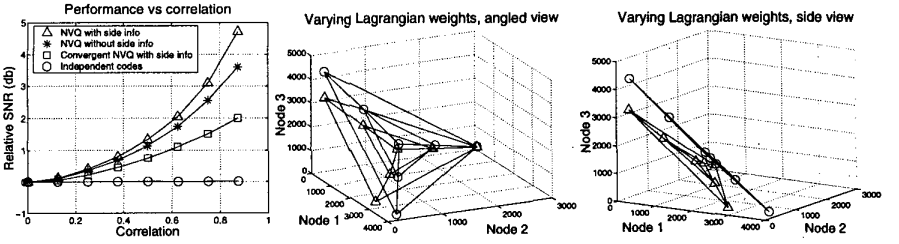

Figure 4: Efficiency of network source coding vs independent coding. (a) Overall performance as a function of correlation. (b),(c) Weighted sum distortion at each node as a function of the Lagrangian parameters (shown from two different angles).

paper) boosts NVQ performance, and that the alterations required for the convergent design algorithm hinder performance.

Using a satellite weather image data set (obtained courtesy of the University of Hawaii and NASA), a performance profile as a function of the Lagrangian weights $\mu_{t,S,r}$ is created. We impose the constraint that all weights $\mu_{t,S,r}$ corresponding to reproductions at the same node are identical and that the sum of all the weights is 1. We sum the weighted distortions of the reproductions for each node and plot these distortions as points in 3-d space as shown in Figures 4(b),(c). Each coordinate represents the weighted distortion at a different node. Varying the Lagrangian weights traces out a surface in this space. The surface corresponding to the NVQs always lies closer to the origin than that of the independently designed VQs, indicating that lower values of weighted distortion are achieved by the NVQs.

6 Summary

In this paper we discuss the practical implementation of an existing fixed-rate network VQ design algorithm and extend the algorithm to allow the messages encoded at a particular node to be used as side information at the decoder of that node. The discussion provides, as special cases, previously unknown optimal fixed-rate VQ design algorithms for Wyner-Ziv and multiple access (multiterminal) systems. Experiments conducted using VQs designed by the new algorithm show that network source coding clearly outperforms separate coding for each transmitter-receiver pair when the sources in the network are correlated.

References

[1] M. Effros. Network source coding. In *2000 Conference on Information Sciences and Systems*, Princeton, New Jersey, March 2000. IEEE.