Slepian-Wolf Coding and Related Problems

Julius Kusuma
kusuma@mit.edu
Massachusetts Institute of Technology
Laboratory for Information and Decision Systems
Cambridge, MA, U.S.A.

October 22, 2001

Abstract

In their landmark paper [6], Slepian and Wolf introduced the idea of using binning to prove the coding theorem for source coding with side information. Wyner and Ziv [7] later showed the rate-distortion function for such a system. In this paper we summarize the recent work on constructing practical codes for source coding with side information using the framework of coset codes [3]. We show promising results where we can get close to the promised performance using well-studied error-correcting codes.

1 Source coding with side information

Slepian and Wolf derived the achievable rate region for the problem of source coding with side information. Suppose an encoder observes X, which is correlated with some observation Y which is already available at the decoder. They showed that there exists a method to optimally compress X to the conditional entropy $H(X|Y)$. Wyner-Ziv then extended this to the lossy case. We begin by describing the encoding and decoding strategy.

1.1 Slepian-Wolf binning arguments

Distributed source coding is the scenario where there are two stochastic observations Y and Z that have a joint probability space $p(Y, Z)$. Two different encoders wish to collaboratively encode their observations in the absence of communication between the two nodes. A central decoder will receive the codewords and will simultaneously decode the observations from both encoders. We now give a formal definition.

Definition: **Slepian-Wolf Distributed source Coding (SWDC)** is the encoding of correlated observations of alphabets $\mathcal{X}$ and $\mathcal{Y}$ by two encoders in the absence of communication between the encoders. The codewords are then transmitted losslessly to a central decoder, which will then decode both observations simultaneously. We define a dummy random variable of alphabet $\mathcal{Z}$ to describe the given joint probability space by $p(x^n, y^n|z^n)$. The observations are said to be memoryless if:

$$p(x^n, y^n) = \prod_{i=1}^{n} p(x_i, y_i) \tag{1}$$

or for x_i independent,

$$p(x^n, y^n|z^n) = \prod_{i=1}^{n} p(x_i, y_i|z_i) \tag{2}$$

The decoder wishes to estimate the dummy random variable Z, together with X and Y using the received messages from the two encoders. Fig. 1 shows the block diagram of the SWDC.

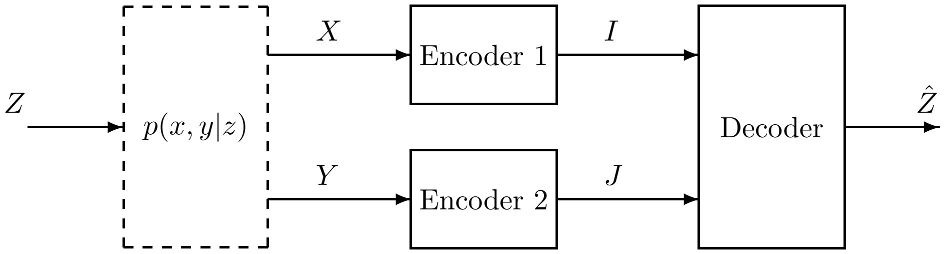


Figure 1: Distributed source coding

STRATEGY: First we describe the strategy for the "corner case". The strategy is to introduce binning such that we get to the part of information about X that is independent to Y. The index of the bin is transmitted, which we denote by U. Thus the discount in the rate: $H(X|Y) = H(X) - I(X;Y)$. Note that we have $p(y|x)$ and the capacity of this channel is $I(X;Y)$. This is called the "asymmetric coding" since we spend different numbers of bits for X and Y. For the "symmetric coding" case, we bin both the space of X and Y, with bin indices U and V transmitted.

Here we make a note of the special case when the “full information” of Y is available at the decoder, i.e. $R_2 = H(Y) + \epsilon$ and only “partial information” of X was sent, i.e. $R_1 = H(X|Y) + \epsilon$ as in Fig. 2. In this case, the rate-distortion bound on X is given as:

$$R^*(d) = \inf I(U;X) - I(U;Y) \tag{3}$$

This is known as the Wyner-Ziv coding theorem. This can be interpreted as SWDC with V partitioned into bins with exactly one member each. It's worth noting that for this special case, one can use any good source coder to allow distributed joint compression of the other correlated observation!

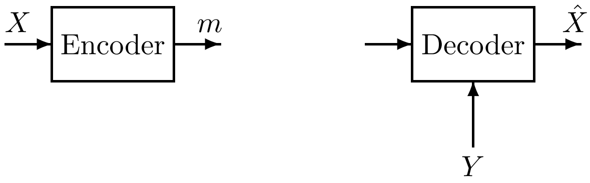


Figure 2: Source coding with side information

1.2 Optimization in the expression

We describe **source coding with side information** when the side information is available at both the encoder and the decoder, and when it's available only at the decoder.

Side information only at the decoder: Wyner and Ziv [8] showed that the rate-distortion function of source coding with side information V only at the decoder is given by:

$$R^*(d) = \inf_{p(u|x)} \inf_{f} I(U;X) - I(U;V) \tag{4}$$

First note that as we consider this to be the corner point of SWDC, we consider Y to be available with high reliability (losslesly) according to the produced representation V. Therefore in the context of mutual information, V and Y can be used interchangeably. Note that the encoder produces U as a function of its observation X, and this process is independent of the side information Y even though X and Y are correlated. This is equal to saying that we insist that $I(Y;U|X) = 0$, or what $U \to X \to Y$ is a Markov chain.

Therefore the infimum is taken over all conditional densities $p(u|x)$ such that $U \to X \to Y$ is a Markov chain and functions $f : \mathcal{U} \times \mathcal{V} \to \hat{\mathcal{X}}$. The decoder then reconstructs $\hat{X} = f(U,Y)$ from the variable U and the side information Y. Because of this, we have a Markov chain $X \to U,Y \to \hat{X}$. U is an auxiliary random variable, and we are interested in the function $\omega : X \to \Re$ that satisfies the distortion constraint:

$$E\left[\rho(X, \hat{X} = f(U,Y))\right] \leq d \tag{5}$$

Because of the Markov chain $U \to X \to Y$, we can write:

$$\begin{aligned} I(U;X) - I(U;Y) &= H(U) - H(U|X) - H(U) + H(U|Y) \\ &= H(U|Y) - H(U|X) \\ &\overset{*}{=} H(U|Y) - H(U|X,Y) \\ &= I(X;U|Y) \end{aligned} \tag{6}$$

where $*$ is due to the Markov property.

Side information at both the encoder and decoder: When the side information is available at both the encoder and the decoder, the rate-distortion function is given by:

$$R_{x|v}(d) = inf_{p(\hat{x}|x,v)} I(X;\hat{X}|V) \tag{7}$$

such that $E\left[\rho(X, f(U,Y))\right] \leq d$.

We are interested in the case where the two capacities are equal. We set $R^*(d) = R_{x|v}(d)$, and see that they are equal if and only if:

$$I(X;U|V) = I(X;\hat{X}|V) \tag{8}$$

Since $\hat{X} = f(U,Y)$, the data processing inequality yields:

$$I(X;U|Y) \geq I(X;\hat{X}|Y) \tag{9}$$

with equality if and only if:

$$I(X;U|\hat{X},Y) = 0 \tag{10}$$

or that $X \to \hat{X}, Y \to U$ is a Markov chain.

1.2.1 Intuition behind source coding with side information

STRATEGY: The strategy is to introduce binning such that we get to the part of information about X that is independent to Y. Thus the discount in the rate: $H(X|Y) = H(X) - I(X;Y)$. Note that we have $p(y|x)$ and the capacity of this channel is $I(X;Y)$.

2 Motivation and a simple construction

We discuss a simple example that illustrated our idea: Consider the following riddle that holds the key to the idea. Suppose X and Y are equiprobable 3-bit binary words that are correlated in the following sense: the Hamming distance between X and Y is no more than one: i.e., X and Y differ in at most one bit-position. If Y is available to both the encoder and decoder, clearly it is wasteful to describe X using 3 bits, as there are only 2 bits of uncertainty between X and Y (there are only 4 possibilities for the modulo-two binary sum of X and Y: $\{000, 001, 010, 100\}$, which can be indexed and sent).

Now what if Y were revealed only to the decoder but not the encoder: could X still be described constructively using only 2 bits of information? A moment's thought reveals that the answer is indeed yes. The solution consists in realizing that since the decoder knows Y, it is wasteful for X to spend any bits in differentiating between $X = 000$ and $X = 111$. Why? The Hamming distance of 3 between these two codewords is sufficiently "large" to enable the decoder to correctly decipher X based on the knowledge of Y which is known to be within Hamming-distance of 1 of the "correct" X. That is, if the decoder knows X to be either $X = 000$ or $X = 111$, it can resolve this uncertainty by checking which of them is closer in Hamming distance to Y and declaring that as the value of X.

Note that the set $\{000, 111\}$ is nothing but a 3-bit repetition code with a Hamming-distance of 3. Likewise, in addition to the set $\{000, 111\}$, consider the following 3 sets for X: $\{100, 011\}$, $\{010, 101\}$, and $\{001, 110\}$. Each of these sets is composed of pairs of words whose Hamming distance is 3. These are just simple variants or cosets of the 3-bit repetition code. While we typically use the set $\{000, 111\}$ as the 3-bit repetition code (0 is encoded as 000, and 1 as 111), it is clear that one could just as well have used any of the other three with the same performance. Further, these 4 sets cover the complete space of all possible binary 3-tuples that X can assume. Thus, instead of describing X by its 3-bit value, all we need to do is to encode which coset X belongs to.

There are 4 equiprobable cosets, and this translates to a cost of 2 bits, just as in the case where Y is known to both encoder and decoder. It turns out that these channel-code cosets are associated with so-called syndromes of the principal underlying channel code (the 3-bit repetition code in the above example). The construction procedure for the encoder is to compute the syndrome of X with respect to the appropriate channel code (the actual code strength has to be matched to level of correlation between X and Y: the lower the correlation, the stronger the required code strength), and transmit this syndrome. The decoding procedure is to identify the closest codeword to Y in the channel code coset associated with the transmitted syndrome, and declare that to be X. The DISCUS acronym follows from exactly this construction.

2.1 Enabling rate-distortion: quantization using cosets

We are interested in the case where X and Y are real-valued random variables. In this article we consider a simple correlation structure between the source and the side information to illustrate the essential concepts. The approach presented here can be extended to capture more elaborate correlation structures. We consider the specific case where the correlation between X and Y is captured as follows: Y is a noisy version of X: i.e., $Y = X + N$, where N is also continuous-valued (defined on the real line $\mathbb{R}$), *i.i.d.*, and independent of X. As before, the setup is that *the decoder alone has access to the Y process*, and the task is to optimally compress the X process. We will consider the case where the X and N are zero-mean Gaussian random variables with known variances: our approach can be generalized to arbitrary distributions for X and N.

2.2 Scalar partitioning example

Consider the following example.

Consider first a simple fixed-length (length-V) scalar quantizer designed for the probability density function of X. Let $V = 8$ for ease of discussion. Let $\nabla = \{r_0, r_1, \ldots, r_{V-1}\}$ be the set of reconstruction levels as shown in the Fig. 3. Note that ∇ partitions the real line into V intervals each associated with one of the reconstruction levels. Thus the source codebook $\mathbb{S} = \nabla$ and $R_s = 3$ bits/sample. If we use this quantizer to encode X, we need to pay the price of 3 bits/sample. We would like to expend less rate (say 1 bit/sample) by exploiting the correlation between the source X and the side information Y while still using the same quantizer. One way to do this is the following. We partition the set ∇ into $M(\leq V)$ cosets. For illustration, let M=2. We group r_0, r_2, r_4 and r_6 into one coset. Similarly r_1, r_3, r_5 and r_7 are grouped into another coset. The channel code $\mathbb{C} = \{r_0, r_2, r_4, r_6\}$ and $R_c = 2$ bits/sample and the rate of transmission is 1 bit/sample. In this illustration we have taken the representation codeword r_i to be the centroid of the disjoint region Γ_i. The encoding can be described as follows:

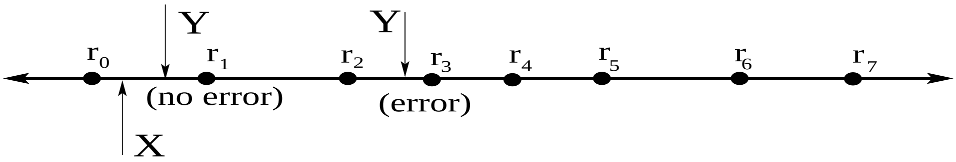


Figure 3: Reconstruction levels of Scalar Quantizer with 8 levels. If Y and X are not close to each other, there is a decoding error.

1 Find the codeword from the set ∇ which is closest (in terms of minimizing the desired distortion measure) to the source sample X. Call this the active codeword.

2 Send the index $U \in \{0, 1, \ldots, M-1\}$ of the coset of $\mathbb{C}$ in $\mathbb{S}$ containing the active codeword.

The decoder deciphers the active codeword by finding the codeword which is closest in some metric to Y in the coset whose index is sent by the encoder. After finding the codeword (say r_k), the decoder estimates X using all the available information. We wish to minimize the expected value of the distortion $\rho(X, \hat{X})$, where $\hat{X}$ is the estimate of X. As discussed before, there is always a finite probability of decoding failure. The probability of decoding failure (see Fig. 3) can be made sufficiently small with more efficient coset constructions. Thus for this case, the source codebook and the channel codebook are both memoryless. For a given rate of transmission R bits/source sample, we choose a scalar quantizer with 2^{R_s} levels and partition it into 2^R cosets each containing 2^{R_c} codewords.

The example above can be extended to more elaborate coset constructions in large dimensional Euclidean spaces with fast algorithms using channel codes as considered in [5]. This framework is based on inducing coset partitioning using channel codes, which we will describe in the next section. Using this, the probability of decoding error can be reduced. Note that in the system the probability of occurrence of the elements in a given coset are not the same. To capture this lack of uniformity we propose an approach based on periodization of the probability density function of the source X. These systems give good gains on scalar sources when compared with the case when the side information is ignored while encoding.

We present one set of simulation results for *i.i.d.* Gaussian sources. We use rate of 1 bit/sample. The distortion and decoding error performance of 4-, 8- and 16- level quantizers are shown in Fig. 10. The optimal rate-distortion bound given by Wyner and Ziv [8] is also shown.

3 Coset coding framework

We give an example of the idea of coset codes based on Hamming codes. Group codes such as Hamming and TCM codes define a subset of all possible codewords called a codebook. The codewords in a codebook are chosen to be maximally separated from each other, for best correlation-noise resilience. For example, in the standard $(7, 4)$ Hamming code, all the codewords in the codebook have syndrome 0. If one applies a one-bit shift to the entire codebook, all the codewords in the new codebook will still have equal syndromes, and still be maximally separated. Thus, by specifying the syndrome, we are selecting ”offsets” from the main codebook centered at 0. Codewords having the same syndrome are said to belong to the same coset.

3.1 Classification and performance limits

The surprising result of [2] is that a large body of Trellis-based codes developed by various researchers fit within the framework of generalized coset codes. Moreover, it was shown that all these codes are based on a coded coset selector and uncoded member selector.

This is shown in Fig. 3.1.

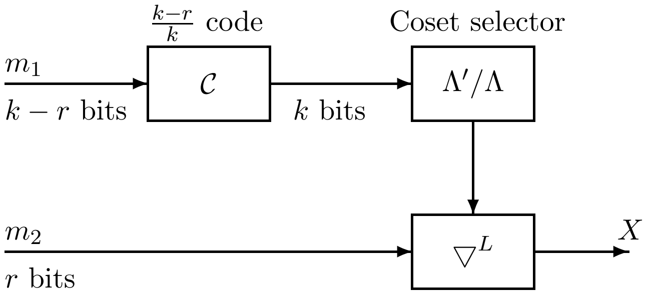


Figure 4: Mapping by set-partitioning: coset codes.

It was shown that the idea is to go to a higher order constellation and applying error-correcting codes where it is needed. What this means is illustrated by the following example: suppose we wish to transmit k bits per symbol time where $k = 3$ in our example. An uncoded approach to this is to use an 8-PSK or 8-QAM constellation. If we wish to use coding, what we do is to first expand the modulation alphabet, for example to 16-QAM. We divide the message stream into two streams: one of 2 bits per symbol, and the other of 1 bit per symbol. The first stream m_1 is first encoded using a rate $\frac{k-r}{k}$-code, which is 2/3 in this case, and the second stream m_2 is left uncoded. The coded stream selects a sequence of cosets Λ/Λ' and the second uncoded stream selects a sequence of members of that coset.

In effect we have expanded the size of the constellation used from 2^k to 2^{k+r}. Note that this scheme maximizes the intra-coset distance at the cost of the inter-coset distance. Therefore we apply codes on error events of inter-coset error. The intra-coset error event is left uncoded.

Forney also showed that it suffices to expand it only by 1 extra bit, i.e. $r = 1$ is sufficient. In fact, the Ungerboeck codes and many other good codes consist of a rate 1/2-code and a coset map on $\mathbb{Z}/4\mathbb{Z}$ for one-dimensional constellations, or $\mathbb{Z}^2/2\mathbb{Z}^2$ for two-dimensional constellations.

In the next part we will further explore this powerful concept and give more explicit examples.

3.2 Geometrically uniform partitioning

Here we give a brief illustration of how to construct practical codes using group-theoretic ideas. This idea strongly parallels the construction used in proving the achievability of the Slepian-Wolf partitioning scheme. Recall that the idea is to partition a codebook into bins, and to select a codeword in the bin selected by the message that is closest to the host signal. After we find such a codeword, the signal X is generated such that the combined transmission $X + S$ is exactly that codeword that we selected. That way we have minimized the perturbation on the host signal. The performance of the datahiding system is limited by the inter-bin distance of the codewords.

The main idea is to generate **linearly independent subcodes** from a main lattice partitioning. We motivate our exploration with the following partitioning example from the $(7, 4)$ Hamming code. Consider partitioning the generator matrix by rows as follows:

$$\mathbf{G} = \begin{bmatrix} 1\,0\,0\,0\,1\,0\,1 \\ 0\,1\,0\,0\,1\,0\,0 \\[4pt] 0\,0\,1\,0\,0\,1\,0 \\ 0\,0\,0\,1\,1\,0\,1 \end{bmatrix}$$

Consider $\mathbf{G_1}$ a new generator matrix consisting only of the first two rows of this generator, and $\mathbf{G_2}$ a new generator matrix consisting of the other two rows. We note three observations:

Geometrically uniform partition: The new generator matrix $\mathbf{G_1}$ is a subgroup that induces a geometrically uniform partition on the original generator.

Label group: The other two rows of this generator matrix $\mathbf{G_2}$ forms a label group of the partition $\mathbf{G}/\mathbf{G_1}$.

Linear independence: We note that the subgroup and the label group, $\mathbf{G_1}$ and $\mathbf{G_2}$, form linearly independent subcodes.

We consider the distance properties of the cosets of the subgroup and the label group.

3.3 Formal group-theoretic constructions of GCC

Before proceeding, we give a formal group-theoretic constructions of Generalized Coset Codes. We will refer to the $(7, 4)$ Hamming code as a real-life example of the constructive elements used here. We begin with a group denoted by S. Let us consider a sequence formed from this group, denoted S^I where I is some index label.

In the Hamming code, let S be $\{0, 1\}$ and I be $\{1, ..., 7\}$. Then S^I is set of all length 7 binary sequences.

4 Several examples of code partitions

Before we get into some practical examples, let us define some codes and give their group interpretations.

4.1 Trellis-based constructions

Here we give several examples of Trellis-based constructions:

1. **16-QAM length-L sequences:** We denote the 16-QAM constellation by $\bigtriangledown$. The set of all length-L sequences which takes a symbol from $\bigtriangledown$ at every sample, $\bigtriangledown^L$, is a geometrically uniform group, taken from $(\mathbb{Z}^2)^L$. The number of elements in this group is 2^{4L}.

2. **16-QAM symbol-space partition:** We can partition $\bigtriangledown$ into two cosets as shown in Fig. 5. Let $\bigtriangledown'$ be such a subgroup that induces that partition. This is a geometrically uniform partition of $\bigtriangledown$. Moreover, the length-L sequences based on this subgroup, $\bigtriangledown'^L$, is a geometrically uniform subgroup of $\bigtriangledown^L$. The size of the quotient group $|\frac{\bigtriangledown^L}{\bigtriangledown'^L}| = 2^L$, which means that the message rate is 1 bit per sample.

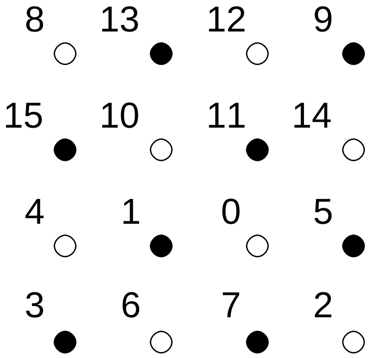


Figure 5: A 16-QAM constellation and the corresponding labels, and partition.

3. **TCM code:** The set of all length-L sequences generated by a rate-3/4 TCM code on $\bigtriangledown^L$ is a geometrically uniform subgroup of $\bigtriangledown^L$. We denote this group by $\mathcal{C}$, described by the generator matrix below:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & D & 1 & 0 \\ 0 & 1 & D^2 & D \end{bmatrix} \tag{11}$$

   The number of elements in this subgroup is 2^{3L}, and the size of the quotient group $|\frac{\bigtriangledown^L}{\mathcal{C}}| = 2^L$. The message rate is again 1 bit per sample.

4. **Subgroup of a TCM code:** By selecting the bottom 2 rows of the generator matrix, we in effect partition the TCM code into a geometrically uniform subcode, which we denote $\mathcal{C}'$. This subcode is described by the following generator matrix:

$$G' = \begin{bmatrix} 0 & D & 1 & 0 \\ 0 & 1 & D^2 & D \end{bmatrix} \tag{12}$$

The number of elements in this subgroup is 2^{2L}. Note that we can induce two different partitions depending on which main lattice we select as the main group. If we select $\bigtriangledown^L$ to be the main group, then the size of the quotient group $|\frac{\bigtriangledown^L}{\mathcal{C}}| = 2^{2L}$, whereas if we select $\mathbb{C}$ to be the main group, then the size of this quotient group $|\frac{\mathcal{C}}{\mathcal{C}'}|$.

Based on the groups that we have defined, we give several practical examples:

1. **Example 1:** We choose Λ to be all length-L sequences from the 16-QAM constellation, and subgroup Λ' chosen such that we have length-L sequences, where at each sample the message will select symbols from the appropriate coset in the picture on Fig. 5. We set $\Lambda = \bigtriangledown$, and $\Lambda' = \bigtriangledown'$.

   The size of Λ is 2^{4L}, and the size of Λ' is 2^{3L}. This partitining admits to a geometrically uniform parition. The size of the message set is 2^L, which means that the message rate is then 1 bit per sample.

   At every sample, the encoder finds the closest element of the coset selected by the message and sends that element at $X + S$. The decoder quantizes the received symbols to the closest element in the main group and estimates the message to be the coset where that element belongs to.

   We refer to this as the *SQ-PAM* code, because the main code is based on quantizing the host signal, and the subcode is based on a PAM code.

2. **Example 2:** Let us illustrate a coding example based on Trellis codes. Let Λ be the same as before: all length-L sequences from the 16-QAM constellation. But now let Λ' be all length-L sequences from a rate-3/4 TCM code on the same 16-QAM constellation.

   It is to see that $|\Lambda| = 2^{4L}$, and $|\frac{\Lambda}{\Lambda'}| = 2^L$. Therefore as we have before, the size of the message set is 2^L and the message rate is 1 bit per sample. We set $\Lambda = \bigtriangledown$ and $\Lambda' = \mathcal{C}$.

   The encoder finds the closest sequence in the subcode $\mathcal{C}$ where at each sample the message bits select the appropriate coset of $\bigtriangledown^L$ induced by $\mathcal{C}$, and finds the sequence closest in the appropriate metric to the host signal. This is exactly Trellis-Coded Quantization. Therefore the encoder can be implemented efficiently as a Viterbi decoder. The decoder quantizes the received sequences to the main group $\bigtriangledown^L$ and finds which coset the symbols belong to at every sample. This is identical to the traditional TCM encoding, and suffers from error propagation.

   We refer to this case as the *SQ-TCM* code, because the main code is based on quantizing the host signal, and the subcode is generated by a TCM code.

   The Trellis description of the two cosets are shown on Fig. 6.

3. **Example 3:** We can also define the main code on a Trellis-based code. Let Λ be a code based on a rate-3/4 TCM code on a 16-QAM constellation. And let Λ' be a code based on a rate-2/3 TCM code on the same 16-QAM constellation.

   We set $\Lambda = \mathcal{C}$ and $\Lambda' = \mathcal{C}'$. In this case $|\Lambda| = 2^{3L}$, and $|\Lambda'| = 2^{2L}$. Therefore $|\frac{\Lambda}{\Lambda'}| = 2^L$.

   The encoder is similar to that of the previous case, but now the decoder is also a Viterbi decoder, with a similar codebook. At the encoder, the message bits select which coset we have to be at each sample, while at the decoder we want to quantize the received signal to the main group, and then find which cosets the symbols belong to at every sample.

   We refer to this case as the *TCQ-TCM* code, because the main code is based on Trellis-quantizing the host signal, and the subcode is generated by a TCM code.

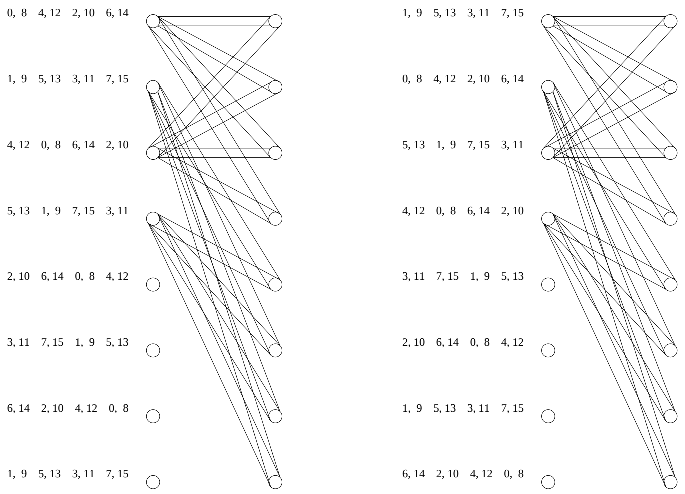


Figure 6: The two trellises formed from the cosets of the code $\mathcal{C}$

4.2 Illustration of decoding

For the symmetric case, we give the following illustration of decoding for distributed compression. The active cosets are denoted by the diamonds and boxes.

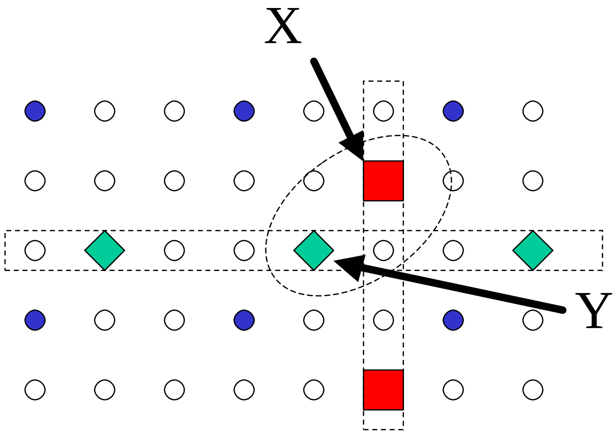


Figure 7: Illustration of coset partitioning for distributed source coding.

5 Optimal quantizers design for the memoryless sources

In [5], Pradhan and Ramchandran considered source coding with side information for the Gaussian case using standard Lloyd-Max design on the PDF of X without considering the statistics of side information S in the quantizer design for X. In this work, we include the side-information in the quantizer design and tackle general memoryless sources. This work was presented in [4]. Note that the ideas here use the framework that we have discussed in the previous sections, and we will use coding-theoretic terminology in our exposition. In the interests of keeping the exposition simple, in this paper, we restrict ourselves to fixed-length (b-bit) scalar quantizers. Extensions to entropy-constrained scalar quantization are straightforward for the "uncoded" case but involved for the "coded" case (more on this later). Thus, in our examples to follow, each parent node communicates its full b bits of sensor data, children send $(b - m)$ bits compressed according to the scheme to be described. We have for this paper restricted ourselves, without loss of generality, to $b = 3$ and $m = 1$: i.e., parents send 3-bit data, while children send 2-bit data.

5.1 Quantization for Gaussian sources

Recall the coset partitioning idea that we develop in a previous section. Consider the asymmetrical coding case of 2 remote encoders and one central decoder. The parent encoder sends full information of its sensor readings to the decoder. The child encoder, without knowledge of the parent encoder's data, sends partial information of its sensor readings in the form of the syndrome (coset index) associated with its quantized representation with respect to a "suitably chosen" code that captures the correlation between the encoders. The coset index alone is ambiguous; the decoder uses the full information from the first reading to estimate the second sensor reading. In designing these coset codes, we form cosets with codeword members as far apart as possible to minimize the probability of misdetection. Given a set of training data and want to design an encoder that takes advantage of the side information available at the decoder. We apply the standard Lloyd-Max design algorithm on the PDF of the sensor reading X.

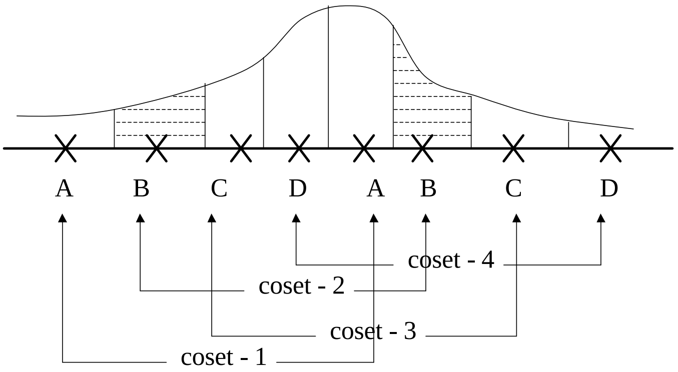


Figure 8: Coset partitioning of an 8-level quantizer into 4 cosets. The compression rate is 3 → 2 bits.

The encoder then sends the index of the coset which the quantized version of X belongs to. This is illustrated in Fig. 8.

5.2 Quantization with periodic extension of the probability distribution

Let us now consider extending this work to general memoryless sources. Our main idea comes from an observation of the source encoding depicted in Fig. 8. Let us consider the bins that correspond to the coset label B. The encoder is not concerned by which quantization bin the observation is quantized to. It is concerned only by which coset it belongs to. Therefore to the encoder there is no difference between which member of the coset of the quantizer. Fig. 8 also suggests that the coset error performance is determined by the minimum distance between elements in a coset. In our example it is 4 bins. Note that this gives a forward-error correction interpretation of the performance: if the side information falls outside the error correcting capability of our code, then we will get a coset error. We specify the observation only up to this minimum distance. We denote it by d^*. Therefore to the quantizer, all the elements in a particular coset are considered jointly. To reflect this, we "periodize" the PDF of X with period d^*:

$$f_X^*(x) = \sum_{i=-\infty}^{+\infty} f_X(x + i.d^*) \tag{13}$$

We illustrate this in Fig. 9 below, We periodize the PDF on the top figure with period d^*, to get the middle figure. We then truncate the PDF as shown in the bottom figure. An optimal Lloyd-Max quantizer design is carried out for this truncated "collapsed" PDF, and this optimal quantizer design is then repeated with period d^* for the original PDF of X.:

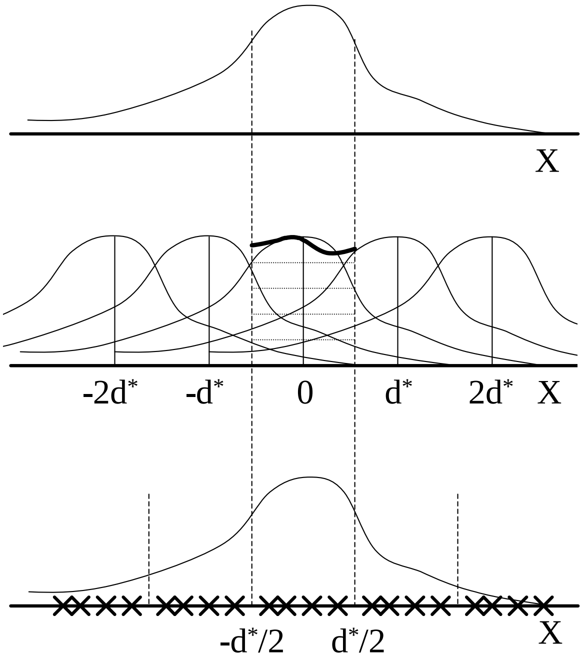


Figure 9: Illustration of PDF periodization.

For training-based design, this periodization is equivalent to appending the sample space by the same samples with different mean bias corresponding to d^*. The quantizer is designed for this "collapsed" PDF. Note that the "collapsed" PDF has lower variance and entropy than the original: *this precisely quantifies the benefit of leveraging the correlated side-information!* To summarize, the design process involves the following steps:

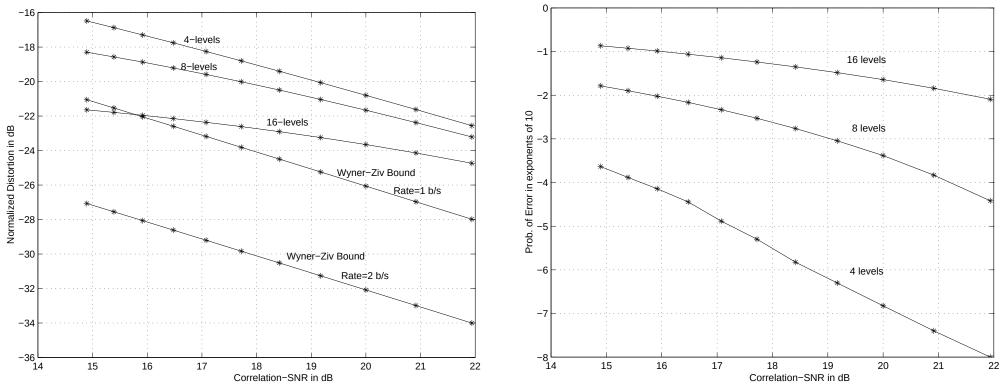
1. **Step 1:** Transform the oridinal PDF of X, $f_X(x)$, by periodizing and truncating in the manner described above. $f_X^*(x)$ has support between $-\frac{d^*}{2}$ and $+\frac{d^*}{2}$.

2. **Step 2:** Do the conventional optimal Lloyd quantizer design on the transformed PDF $f^*(x)$ from Step 1. Recall that this involves iteratively optimizing the reconstruction points $\{q_i\}$ (centroid condition) and quantizer intervals $\{t_i\}$ (nearest neighbor condition) until local convergence.

3. **Step 3:** "Periodize" the quantizer design from Step 2 (i.e., the $\{q_i\}$'s and $\{t_i\}$'s) with period d^* and apply it to the original PDF of X, $f_X(x)$.

5.3 Simulation Results

We present one set of simulation results for *i.i.d.* Gaussian sources. We use rate of 1 bit/sample. The distortion and decoding error performance of 4-, 8- and 16- level quantizers are shown in Fig. 10. The optimal rate-distortion bound given by Wyner and Ziv [8] is also shown.



(a) (b)

Figure 10: (a) Normalized Distortion versus correlation-SNR: The distortion performance improves as we increase the number of levels in the root quantizer. (b)Probability of decoding failure versus correlation-SNR for different quantizers: the probability of decoding error increases as we increase the size of the root quantizer.

The larger the value of d^*, the more robust the system is to variance between X and S, and hence the lower the coset error rate. However, larger d^* translates to coarser quantization and hence larger quantization error for each sample. There is an optimal d^* which gives the best tradeoff between coset error and quantization error. This is illustrated in Fig. 12. The results are intuitive: different Correlation SNRs (CSNRs) give different optimal d^* values. Higher SNR means that we should pick d^* to be smaller, and gives better quantization performance. As d^* is increased, then the CSNR difference becomes less apparent because now most of the distortion is incurred due to quantization. For a given rate and CSNR constraint, we should optimize d^* to minimize the distortion. The simulation results of the coset error probability are consistent with the results of TCM for channel coding. We obtained a coding gain of $2-3$ dB for high CSNR, and the increased intra-coset distance is apparent. Note that for smaller values of d^*, the periodization of the PDF means that the new PDF is very close to being uniform. Therefore the application of Entropy-Constrained Quantizer Design (ECSQ) [1] cannot further decrease the compression rate.

6 Discussions

We have proposed a methodology for reducing the overall bandwidth required to send data in a sensor network based on correlation grouping and no inter-node communication. The reduced information can be recovered at the decoder with the given side information of the parent. We have shown that our methodology is suitable across a variety of quantization levels, compression rates, and SNR levels

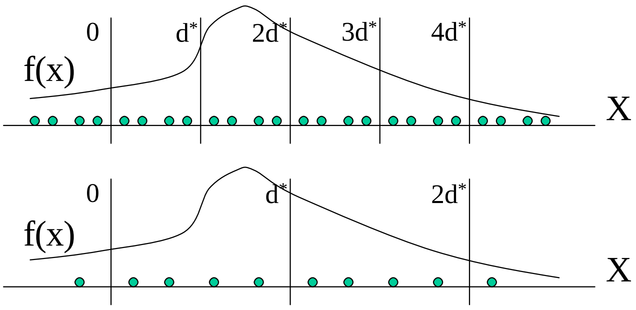


Figure 11: The choice of d^* gives the tradeoff between quantization error and coset error.

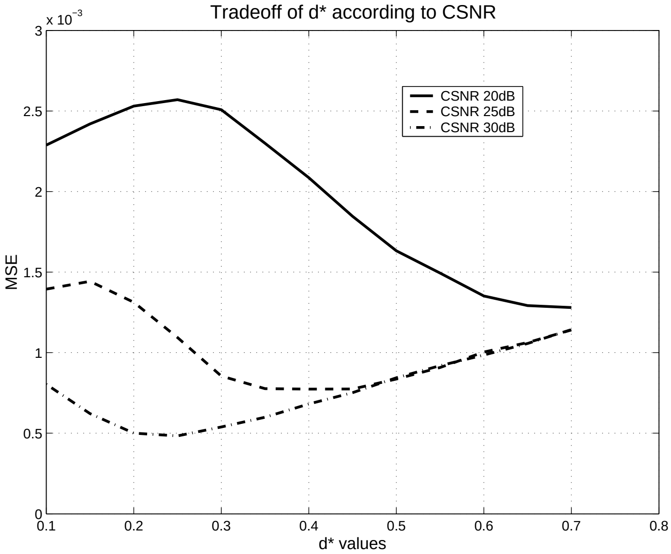


Figure 12: Tradeoff between MSE versus d^* for the coded case: different Correlation SNRs (CSNRs) give different optimal d^* values.

in the sensor network. Furthermore, the simplicity of the coding methods presented allow for easy implementation even with the limited processing capabilities of sensor nodes. The methodology for a single child node is directly extensible to plural children. In this setting, we have a single parent data stream S and many child data streams $X_1, X_2, , X_n$. We assume the same correlation noise N for all parent-child pairs, i.e. $X_i = S + N_i$ for all nodes i. Each child node data stream is encoded as discussed and disambiguated at the decoder using the parent data as side information. As such, the parent node provides side information for an entire cluster of children. As the cluster size grows, the average rate in the network will approach the compressed rate of the children. Ideally the number of clusters could reflect the design parameters and the variability of the environment. If the field is relatively static compared to the sensor node density (for example, the temperature throughout a flat field), we would like to have a small number of parent nodes to minimize the overall number of bits transmitted by the network. In contrast, a highly-variable situation necessitates the formation of several clusters to exploit any correlation whatsoever.

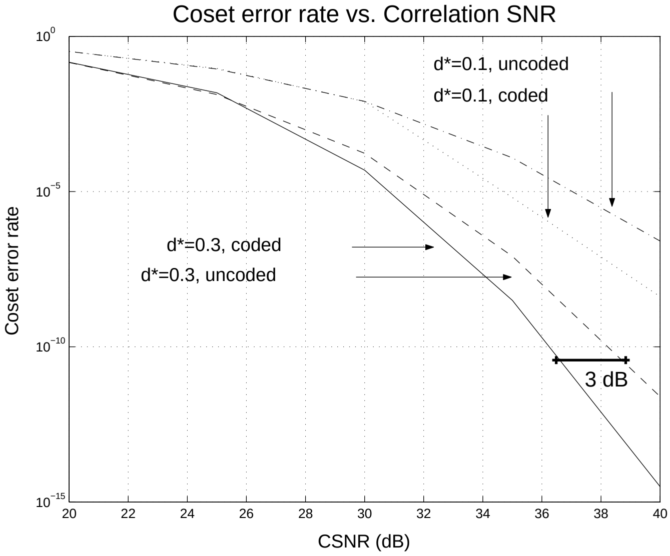


Figure 13: Coset error rate for X Gaussian.

It might also be desirable to rotate the role of the parent node through a cluster. Assuming some temporal correlation, the estimated child data could be compared to full parent data at a given node. If significant discrepancies were observed, the value of d* would be increased for this node. Alternatively, one could use variable length codes within a cluster: nodes closer to the parent in data space would send shorter codes than those farther. A goal of a more complete system would be to incorporate our design goals into a clustering algorithm to achieve the optimal reduction in overall network communication requirements.

References

[1] N. Farvardin and J. W. Modestino. Optimum quantizer performance for a class of non-gaussian memoryless sources. *IEEE Transactions on Information Theory*, IT-30(3):485–497, May 1984.

[2] G. D. Forney. Coset codes-I: Introduction and geometrical classification. *IEEE Transactions on Information Theory*, IT-34:1123–1151, September 1988.

[3] G. D. Forney. Geometrically uniform codes. *IEEE Transactions on Information Theory*, 37:1241–1260, 1991.

[4] J. Kusuma, L. Doherty, and K. Ramchandran. Distributed compression for sensor networks. In *IEEE Conference on Image Processing*, Thessaloniki, Greece, October 2001.

[5] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): Design and construction. In *Proc. of DCC*, Snowbird, UT, 1999.

[6] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, IT-19:471–480, July 1973.

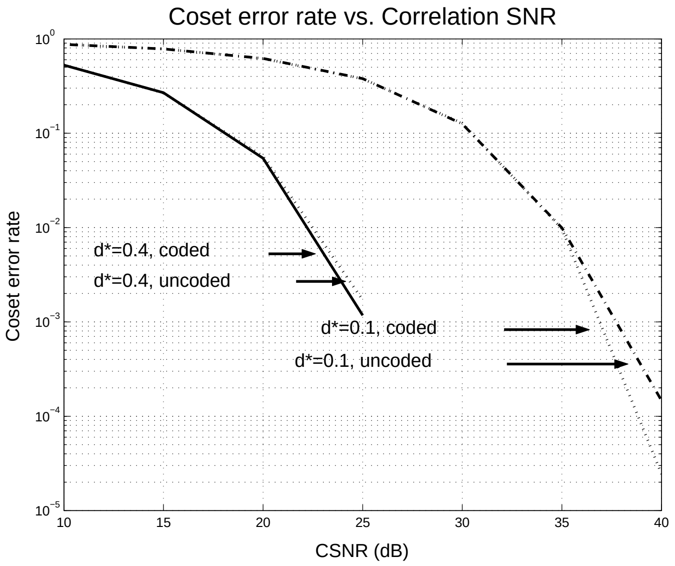


Figure 14: Coset error rate for X uniformly distributed.

[7] A. D. Wyner. Recent results in the shannon theory. *IEEE Transactions on Information Theory*, IT-20:2–10, January 1974.

[8] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, IT-22:1–10, January 1976.