The Non-existence of Length-5 Perfect Slepian-Wolf Codes of Three Sources

Samuel Cheng and Rick Ma

***Abstract*—We consider Slepian-Wolf coding of multiple sources and extend the packing bound and the notion of perfect code from conventional channel coding to SW coding with more than two sources. Moreover, we show that there does not exist perfect Slepian-Wolf code of length-5 for three sources.**

***Index Terms*—Slepian-Wolf coding, distributed source coding, perfect code**

I. Introduction

Despite the renewed interest of practical implementation of Slepian-Wolf (SW) coding [1], most work is restricted to the discussion of two sources [2]–[6] except very few exceptions [7], [8]. In this paper, we consider SW coding of multiple sources. In particular, we will show that a perfect length-5 SW codes of three sources does not exist.

SW coding refers to lossless distributed compression of correlated sources. Consider N correlated sources $X_1, X_2, \cdots, X_N$. Assuming that encoding can only be performed separately that N encoders can see only one of the N sources but the compressed sources are transmitted to a base station and decompressed jointly. To the surprise to many researchers of their time, Slepian and Wolf showed that it is possible to have no loss in sum rate under this constrained situation [1]. That is, at least in theory, it is possible to recover the source losslessly at the base station even though the sum rate is barely above the joint entropy $H(X_1, X_2, \cdots, X_N)$.

Wyner is the first who realized that by taking computed syndromes as the compressed sources, error-correcting parity check codes can be used to implement SW coding [9]. The approach was rediscovered and popularized by Pradhan *et al.* more than two decades later [2], where the scheme is restricted to two correlated sources with one of them treated as side information. In this paper, we generalize this idea to SW coding for any number of correlated sources and extend the packing bound and the notion of perfect code from conventional channel coding to SW coding with more than two sources. In particular, we will show that while perfect length-5 SW codes of 3 sources are suggested from the packing bound, such code does not exist.

II. General Syndrome based SW Coding

We will start with a general definition of syndrome based SW codes as follows.

Definition 1 (Syndrome based SW code)**.** A rate $(r_1, r_2, \cdots, r_N)$ *syndrome based SW code* for N correlated length-n sources contains N parity check matrices $H_1, H_2, \cdots, H_N$ of sizes $m_1 \times n, m_2 \times n, \cdots, m_N \times n$, where $r_i = m_i/n$ for $i = 1, 2, \cdots, N$.

- Encoding: The i^{th} encoder compresses length-n input $\mathbf{x}_i$ into $\mathbf{y}_i = H_i\mathbf{x}_i$ and transmit the compressed m_i bits (with compression rate $r_i = m_i/n$) to the base station
- Decoding: Upon receiving all $\mathbf{y}_i$, the base station decodes all sources by outputting a most probable $\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \cdots, \hat{\mathbf{x}}_N$ that satisfies $H_i\hat{\mathbf{x}}_i = \mathbf{y}_i, i = 1, 2, \cdots, N$.

The definitions above include all prior syndrome SW coding approaches as special cases. For example, the syndrome based asymmetric SW setup is a special case of our syndrome based SW code when $N = 2$ and one of the parity check matrices is set to be an identity matrix.

Definitions 1 do not explain how decoding is performed. Indeed, the SW code defined above can be overly optimistic that no decoding algorithm will be able to recover the sources losslessly. Of course, the required rates will depend on the correlation among the sources. It is expected that the more correlated the sources are and the lower the rates r_i are needed (higher compression is possible). In general, the entire statistics of the sources are completely captured by the joint probability $p(x_1, x_2, \cdots, x_N)$ if we further restrict our sources to be memoryless.

To simplify the language, we will call a possible input to a SW code, i.e., any N-tuple of length-n discrete vectors, as an *(N, n)-configuration*, or simple a *configuration*. Moreover, while a configuration is really an N-tuple of binary vectors, without introducing much confusion, we will also refer a configuration as a *code vector*. More precisely, we have the following definitions.

Definition 2 (Code vector)**.** For a syndrome based SW code defined by $H_1, H_2, \cdots, H_N$, we call N-tuple of length-n binary vectors a configuration $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ as a *code vector with a syndrome* $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_N$ if $H_i\mathbf{x}_i = \mathbf{s}_i, i = 1, \cdots, N$.

Definition 3 (Code word)**.** In particular, we call a code vector with the all-zero syndrome $\overbrace{\mathbf{0}, \cdots, \mathbf{0}}^{N}$, $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$, as a *codeword.*

Further, let us define a *compressible set* and a *compressible configuration* in the following sense.

Definition 4 (Compressible set and compressible configuration)**.** The *compressible set* of a syndrome based SW code contains all configurations that can be recovered losslessly by the SW decoder. And we call a configuration to be *compressible* by the given SW code if it lies within the compressible set.

The following propositions are apparent from the definition.

Proposition 1. *All compressible vectors have different syndromes from the others.*

Proposition 2. *The size of compressible set equal to $2^{\sum_{i=1}^N m_i}$.*

Proof: Each compressible vector is identified by a unique syndrome, and there are $2^{\sum_{i=1}^N m_i}$ syndromes in total. ■

A syndrome based SW code is really a natural extension of linear block codes. We can see that the "linearity" of the code is preserved: if both $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ and $\mathbf{x}'_1, \mathbf{x}'_2, \cdots, \mathbf{x}'_N$ are codewords, the sum $\mathbf{x}_1 + \mathbf{x}'_1, \mathbf{x}_2 + \mathbf{x}'_2, \cdots, \mathbf{x}_N + \mathbf{x}'_N$ is also a codeword. Further, we have the following almost trivial lemma.

Lemma 1. *If both $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ and $\mathbf{x}'_1, \mathbf{x}'_2, \cdots, \mathbf{x}'_N$ are code vectors with the same syndrome and $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \cdots, \tilde{\mathbf{x}}_N$ is a code vector with syndrome $\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \cdots, \tilde{\mathbf{s}}_N$, then $\tilde{\mathbf{x}}_1 + \mathbf{x}'_1 - \mathbf{x}_1, \tilde{\mathbf{x}}_2 + \mathbf{x}'_2 - \mathbf{x}_2, \cdots, \tilde{\mathbf{x}}_N + \mathbf{x}'_N - \mathbf{x}_N$ is a code vector of syndrome $\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \cdots, \tilde{\mathbf{s}}_N$.*

Proof: Since $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ and $\mathbf{x}'_1, \mathbf{x}'_2, \cdots, \mathbf{x}'_N$ have the same syndrome, $H_i \mathbf{x}_i = H_i \mathbf{x}'_i, \forall i$. Therefore, $H_i(\tilde{\mathbf{x}}_i + \mathbf{x}'_i - \mathbf{x}_i) = H_i \tilde{\mathbf{x}} = \tilde{\mathbf{s}}_i$. ■

Despite the simplicity of Lemma 1, it is very important as it implies that every code vector sees the same code distribution (distribution of neighbors with the same syndrome) as any other code vector. As the effectiveness of a code will depend on the correlation model of the source, before proceeding further, we will give a precise symmetry definition for a correlation model considered in this paper[1].

Definition 5 (Symmetric correlation)**.** We call a correlation model specified by the joint probability $p(x_1, x_2, \cdots, x_N)$ as *symmetric* if $p(x_1, x_2, \cdots, x_N) = p_{c_1(x_1, x_2, \cdots, x_N)}$, where $c_1(x_1, x_2, \cdots, x_N) = \sum_{i=1}^N \iota(x_i = 1)$ counts the number of ones in $x_1, \cdots, x_N$ and $\iota(\cdots)$ is the indicator function that is equal to one if the argument is true and zero otherwise.

Definition 6 (Strictly symmetric correlation)**.** A symmetric correlation model is strictly symmetric if $p_i = p_{N-i}$, where p_i is defined in the previous definition.

Strictly symmetric condition essentially ensures that each source is "equally important" statistically and the prior probabilities satisfiies $p_X(0) = p_X(1)$. We will focus ourselves to strictly symmetric case from now on. To further simplify the language, let us call $x_1, x_2, \cdots, x_N$ as a *type-k* ($k \leq \lfloor N/2 \rfloor$) correlation if $c_1(x_1, x_2, \cdots, x_N) = k$ or $c_1(x_1, x_2, \cdots, x_N) = N - k$. For a strictly symmetric source, we will define the profile of a configuration as the empirical distribution of different correlation type in the configuration. And immediately, we have a simple lemma that is self-evident.

Definition 7 (Profile)**.** A profile of an (N, n)-configuration for a strictly symmetric source is a length-$\lfloor N/2 \rfloor + 1$ vector that the k^{th} component equal to the number of type k^{th} correlations in the configuration.

Lemma 2. *If the correlation of a source is strictly symmetric, any two configurations of the same profile will have same probability of occurrence.*

Corollary 1. *If a source is strictly symmetrically correlated, then a configuration χ will have the same probability of that of $\chi + \chi_0$, where χ_0 is a configuration with type-0 correlation only.*

Proof: Note that both χ and $\chi + \chi_0$ have the same profile and thus the probabilities of occurrence are the same from Lemma 2. ■

For a typical application such as sensor networks, $x_1, \cdots, x_N$ are highly correlated and we expect the probability of type-k correlation decreases as k increases up to $N/2$. Therefore, we may want to design a code such that it can compress all configurations with all type-0 correlation except up to a certain number of type-k' correlation, $0 < k' \leq k$. Through simple counting, we have a sphere packing bound analogous to that of conventional error correcting codes.

Lemma 3. *If a code $\mathcal{C}$ can compress all configurations with t or less type-k' correlations, $0 < k' \leq k < \lceil N/2 \rceil$-1, (and $n - t$ or more type-0 correlations,) then $2^{Nn} \geq |\mathcal{C}| 2^n \sum_{t'=0}^t \left[\binom{n}{t'} \left[\sum_{k'=1}^k \binom{N}{k'} \right]^{t'} \right]$, where $|\mathcal{C}|$ denotes the cardinality of $\mathcal{C}$ equal to the total number of codewords. For the special case when N is even and $k = N/2$, if a code $\mathcal{C}$ can compress configurations with up to t type-k' correlations, $k' \leq k$, (and $n - t$ or more type-0 correlations,) then $2^{Nn} \geq |\mathcal{C}| 2^n \sum_{t'=0}^t \left[\binom{n}{t'} \left[2^{N-1} - 1 \right]^{t'} \right]$.*

Proof: The number of configurations with $n - t$ type-0 correlations and t type-k' correlations, $0 < k' \leq k$, is $\binom{n}{t} 2^{n-t} \left[\sum_{k'=1}^k \binom{N}{k'} + \binom{N}{N-k'} \right]^t$. Therefore, the total number of code vectors with up to t type-k' correlations, $k' \leq k$, is $2^n \sum_{t'=0}^t \left[\binom{n}{t'} \left[\sum_{k'=1}^k \binom{N}{k'} \right]^{t'} \right]$ and by Proposition 2, it has to be less than the number representable by the total number of syndromes, which is $2^{Nn}/|\mathcal{C}|$. The proof for special cases when N is even and $k = N/2$ are similar and hence omitted. ■

Analogous to conventional error correcting codes, a *perfect code* is defined as a code that achieves the packing bound specified by Lemma 3.

Example 1. For $N = 2$ and $n = 7$, the codes that can compress all configurations up to 1 bit difference (i.e., all type-0 correlations except no more than one type-1 correlation) satisfies $2^{2n} \geq |\mathcal{C}| 2^n \sum_{t'=0}^1 \binom{n}{t'} = |\mathcal{C}| 2^n (1 + n)$. That means that the code should at least has $\log_2(2^{2n}/|\mathcal{C}|) \geq \log_2(2^n(1 + n)) = 10$ syndrome bits. Such perfect codes $\mathbf{y}$ can be constructed based on $(7, 4)$-Hamming code, where the 10 syndrome bits that can be leveraged over the two encoders [3], [8]. For example, $H_1 = \begin{bmatrix} 1000000 \\ 0100000 \\ 0010100 \\ 0011010 \\ 0011001 \end{bmatrix}$ and $H_2 = \begin{bmatrix} 0010000 \\ 0001000 \\ 1100100 \\ 0100010 \\ 1000001 \end{bmatrix}$ when both encoders compress the source symmetrically from 7 bits to 5 bits (and hence 10 bits in total).

Example 2. For $N = 3$ and $n = 5$, Lemma 3

[1] Note that the symmetry of a correlation defined here is quite different from that of a channel [10].

concludes that a code that can compress all configurations up to one type-1 correlation needs to satisfy $2^{3n} \geq |\mathcal{C}|2^n \sum_{t'=0}^{1} \left[\binom{n}{t'}\left[\sum_{k'=1}^{1}\binom{3}{k'}\right]^{t'}\right] = |\mathcal{C}|2^n(1+3n)$. Therefore, the total number of syndrome bits needed $\log_2(2^{3n}/|\mathcal{C}|) \geq \log_2(2^n(1+3n)) = \log_2(2^9) = 9$ bits. It suggests that a perfect code of length 5 that compresses 15 input bits into 9 bits in total is possible. However, as will be shown in the next section, such perfect code does not exist.

III. Main result

Proposition 3. *There does not exist length-*5 *perfect SW code of three sources.*

Proof: A length-5 SW code that can potentially compress all configurations up to one type-1 correlation into 9 bits in total is suggested in Example 2. Let us denote S as the set containing all these configurations.

Let H_1, H_2, H_3 be the three parity check matrices. Our strategy is to limit the null spaces of them by the fact that all elements in S need to have distinct syndromes. The limitation will eventually kill the possibility of the existence of H_1, H_2, H_3.

Denote the null set of a matrix A as null$(A) = \{\mathbf{u}|A\mathbf{u} = \mathbf{0}\}$, where $\mathbf{0}$ is an all zero vector. Further, denote $\mathbf{e}_i$ as the length-5 binary column vector that has i^{th} component equal to 1 and the rest of its components equal to zero.

We may assume the number of row in H_1 is smaller than or equal to the other's. In other words, H_1 has at most 3 rows. Hence null(H_1) has at least two degrees of freedom. Regardless the values of H_2 and H_3, null(H_1) cannot contain any $\mathbf{e}_i$. Otherwise, both $(\mathbf{e}_i, \mathbf{0}, \mathbf{0})$ and $(\mathbf{0}, \mathbf{0}, \mathbf{0})$ that are in S will get the same outputs $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$. Similarly, null$(H_1)$ cannot contain $\mathbf{e}_i + \mathbf{e}_j$ neither, otherwise both $(\mathbf{e}_i, \mathbf{0}, \mathbf{0})$ and $(\mathbf{e}_j, \mathbf{0}, \mathbf{0})$ (in S) will get the same outputs because $H_1\mathbf{e}_i = H_1\mathbf{e}_j$. Thus null$(H_1)$ can only be span$(\mathbf{e}_i+\mathbf{e}_j+\mathbf{e}_k, \mathbf{e}_i+\mathbf{e}_m+\mathbf{e}_n)$, where the letters i, j, k, m, n are different to each others. i.e. null$(H_1) = \{\mathbf{0}, \mathbf{e}_i + \mathbf{e}_j + \mathbf{e}_k, \mathbf{e}_i+\mathbf{e}_m+\mathbf{e}_n, \mathbf{e}_j+\mathbf{e}_k+\mathbf{e}_m+\mathbf{e}_n\}$ for some i, j, k, m, n. Other structures such as higher dimension will contain forbidden elements. As the dimension of the null spaces of 1×5 and 2×5 matrices are all greater than 2, H_1 has to have at least 3 rows. Thus both H_2 and H_3 also have three rows. It excludes the possibility of perfect asymmetric SW codes (at rate $[2/5, 3/5, 4/5]$, for example). So we can focus only on the symmetric case from now on.

From the above discussion, null(H_1) has to contain $\mathbf{0}$, two "3**e**" vectors and one "4**e**" vector, and no more. Similarly, null(H_2) and null(H_3) get the same structure.

Without lose of generality, we can write null$(H_1) = \{\mathbf{0}, \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_1 + \mathbf{e}_4 + \mathbf{e}_5, \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5\}$. Suppose null$(H_2)$ contain $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$, then of course null$(H_3)$ cannot contain $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$. Otherwise, $(\mathbf{0}, \mathbf{0}, \mathbf{0}) \in S$ and $(\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3) \in S$ get the same output $(\mathbf{0}, \mathbf{0}, \mathbf{0})$. But null$(H_3)$ cannot contain $\mathbf{e}_i + \mathbf{e}_j + \mathbf{e}_k$, $i, j \in \{1, 2, 3\}$; $k \in \{4, 5\}$ $(i \neq j)$. Otherwise $(\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_i + \mathbf{e}_j) \in S$ and $(\mathbf{0}, \mathbf{0}, \mathbf{e}_k) \in S$ share the same output as well. So the "3**e**" vectors of null(H_3) can only be two of $\mathbf{e}_1+\mathbf{e}_4+\mathbf{e}_5$, $\mathbf{e}_2+\mathbf{e}_4+\mathbf{e}_5$, and $\mathbf{e}_3+\mathbf{e}_4+\mathbf{e}_5$. Unfortunately, any pair of them sum up to a "2**e**" vector instead of a "4**e**" vector. Therefore, there cannot be a common "3**e**" vector shared between any pair of the null spaces of H_1, H_2, and H_3.

Now, suppose null(H_2) contains the same "4**e**" vector $\mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5$ as null(H_1) does. Then null(H_3) cannot contain any "4**e**" vector. Let the "4**e**" vector of null(H_3) be $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5 - \mathbf{e}_j$, $j \in \{2, 3, 4, 5\}$. Then $(\mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5, \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5, [1, 1, 1, 1, 1]^T)$ and $(\mathbf{0}, \mathbf{0}, \mathbf{e}_j)$ shares the same syndrome. Thus, any pair of null spaces of H_1, H_2, and H_3 cannot share a common "4**e**" vector as well.

Hence, without loss of generality, we can write null$(H_2) = \{\mathbf{0}, \mathbf{e}_2 + \mathbf{e}_1 + \mathbf{e}_4, \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_5, \mathbf{e}_1 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5\}$. Then, there are only three different possibilities for the "4**e**" vector of null(H_3):

1) $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_4 + \mathbf{e}_5$;
2) $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_5$;
3) $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4$.

Case 1 does not work because $(\mathbf{e}_1 + \mathbf{e}_4 + \mathbf{e}_5, \mathbf{e}_1 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5, \mathbf{e}_1 + \mathbf{e}_4 + \mathbf{e}_5) \in S$ and $(\mathbf{0}, \mathbf{0}, \mathbf{e}_2) \in S$ shares the same syndrome.

Case 2 does not work neither because $(\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_5, \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_5) \in S$ and $(\mathbf{0}, \mathbf{e}_1, \mathbf{0}) \in S$ share the same syndrome.

Finally, case 3 fails as well since $(\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4, \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4) \in S$ and $(\mathbf{0}, \mathbf{e}_3, \mathbf{0}) \in S$ share the same syndrome. ■

IV. Conclusion

We generalized the syndrome based approach of SW coding to more than two sources and showed the packing bound as an extension of that in channel coding. We naturally extended the definition of perfect code as a SW code that satisfies the SW packing bound and pointed out SW perfect code example originated from [3]. While the packing bound suggests that there may exist a length-5 perfect SW code for 3 sources, we showed that that no such SW code exists.

References

[1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, Jul. 1973.

[2] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): design and construction," in *Proc. DCC*, 1999, pp. 158–167.

[3] D. Schonberg, K. Ramchandran, and S. S. Pradhan, "Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources," in *Data Compression Conference, 2004. Proceedings. DCC 2004*, 2004, pp. 292–301.

[4] B. Rimoldi and R. Urbanke, "Asynchronous Slepian-Wolf coding via source-splitting," in *ISIT'97*, Ulm, Germany, 1997, p. 271.

[5] J. Garcia-Frias and Y. Zhao, "Near-Shannon/Slepian-Wolf performance for unknown correlated sources over

AWGN channels," *Communications, IEEE Transactions on*, vol. 53, no. 4, pp. 555–559, 2005.

[6] J. Chen, D.-k. He, A. Jagmohan, and L. A. Lastras-Montano, "On the reliability function of variable-rate Slepian-Wolf coding," in *45th Annual Allerton Conference*, Urbana-Champaign, IL, 2007.

[7] A. Liveris, C. Lan, K. Narayanan, Z. Xiong, and C. Georghiades, "Slepian-Wolf coding of three binary sources using LDPC codes," in *Proc. Intl. Symp. Turbo Codes and Related Topics*, 2003.

[8] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiades, "On code design for the Slepian-Wolf problem and lossless multiterminal networks," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1495–1507, 2006.

[9] A. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inform. Theory*, vol. 20, pp. 2–10, Jan. 1974.

[10] T. Cover and J. Thomas, *Elements of Information Theory*, 2nd ed. New York: Wiley, 2006.