

Density Evolution Technique for LDPC Codes in Slepian-Wolf Coding of Nonuniform Sources

Raghunadh K Bhattar
SNAA
SAC-ISRO
Ahmedabad, India

K R Ramakrishnan
EE Dept
IISc
Bangalore, India

K S Dasgupta
SNPA
SAC-ISRO
Ahmedabad, India

ABSTRACT

This paper attempts to examine the optimality of LDPC codes for compression of nonuniform source with Slepian-Wolf coding using density evolution technique. The primary goal is to evaluate the performance of LDPC codes with reference to turbo codes (in SF-ISF setup). The appreciable difference between LDPC and turbo codes is also discussed in this paper. The threshold values obtained from the density evolution technique indicate that the conditional entropy $H(X/Y)$ is nearly constant with source distribution. This feature is useful in calculating the threshold values for any given source distribution analytically. This special feature is true for only LDPC codes. Several well known LDPC codes, both regular and irregular are critically analyzed using density evolution technique. This analysis reveals that the capacity approaching LDPC codes with respect to error correction codes do indeed approach the Slepian-Wolf bound for nonuniform sources as well. The threshold values show that the nonuniform source can be compressed to near about 0.01bits/sample away from Slepian-Wolf bound even for highly decorrelated side information.

General Terms

Signal Processing, Distributed Source Coding

Keywords

LDPC, Nonuniform Source, Slepian-Wolf Coding, Density Evolution, DSC

1. INTRODUCTION

The Slepian-Wolf theorem [1] is generally applied to uniformly distributed binary sources to exploit the correlation between two sources. This is also known as distributed source coding (DSC) or compression of binary sources with side information. Predominantly the main aim of the distributed source coding is to exploit the correlation between two sources which are physically separated and do not communicate with each other, but acquire (or sense) correlated data. The classic instances of these type situations arise in sensor networks, video surveillance applications etc., As the sources cannot communicate with each other, there is no way to exploit the correlation at the encoder side. However, the decoder has access to both sources, it would be possible for the decoder to make use of this correlation. This problem was theoretically addressed way back in 1970s [1], [2]. Thirty years after the discovery of Slepian-Wolf theorem, practical distributed source coding schemes using channel codes have appeared in literature [3]. The encoding of channel codes is simpler while decoding is more complicated. This paradigm of source coding with channel codes results in low complexity encoder and

transfers most of the computational load to decoder. (In traditional source coding, generally source encoder is several times more complex than the decoder). This exchange of computational complexities find applications in low complexity video and image encoding such as wireless communications, satellite imaging etc.,

According to Slepian-Wolf theorem, two binary sources X and Y can be compressed lossless with the bit rates equal to R_x and R_y respectively as long as the below conditions are satisfied.

$$\begin{aligned} R_x &\geq H(X/Y) \\ R_y &\geq H(Y/X) \\ R_x + R_y &\geq H(X, Y) \end{aligned} \quad (1)$$

where $H(X/Y)$ and $H(Y/X)$ are conditional entropies and $H(X, Y)$ is the joint entropy. It is generally expected that the binary data to Slepian-Wolf encoder is correlated and the binary sources themselves are non-redundant (the probability of occurrence of '0' and '1' is same). Hence the research efforts in DSC are concentrated around the exploitation of correlation between the sources rather than the source redundancy. But in real life, this may not always be true and input to Slepian-Wolf encoder may contain some redundant data. For instance, binary data sources, such as raw facsimile images, textual data and engineering drawings are highly nonuniform and contain large amount of redundancy. For example, uncompressed binary images such as facsimile images have the source probability around $P(X = 0) = 0.96$ [4]. In such situations the source redundancy need to be removed by employing some binary data compression algorithms before the Slepian-Wolf coding is applied. This may spoil the source correlation with side information. This problem may be tackled in two ways: i) by finding a source code (or a channel code) which can exploit both source redundancy and correlation with side information, ii) by applying entropy coding techniques on the Slepian-Wolf compressed bit stream. A combined scheme was applied in [5] using turbo-binning followed by variable-length syndrome sequences. Besides turbo codes, distributed arithmetic codes are also applied for nonuniform sources in [6], [7].

In [5] the authors tried to optimize turbo codes [8] for nonuniform sources and found that the optimized turbo codes are not adequate for approaching Slepian-Wolf bound and there is a considerable gap (0.3574 from SW bound) between achievable compression and Slepian-Wolf bound.

In order to reduce the gap they have proposed a variable length coding (entropy coding) on the turbo bin-indices. This approach

still leaves the gap to capacity as much as 0.1444 bits/symbol for highly nonuniform sources. The authors of [5] have felt that a one step approach to compute the variable length syndromes/bin-indices is difficult. Motivated by this shortcoming in turbo codes, we have investigated the effectiveness of LDPC codes for nonuniform sources.

This paper demonstrates the suitability and optimality of LDPC syndrome codes for distributed source coding of nonuniform sources using *density evolution*. Both regular and irregular LDPC codes are evaluated and threshold values are determined for different source distributions.

The paper is organized as follows. The encoding and decoding of nonuniform sources are presented briefly in Section 2. Section 3 describes the density evolution for nonuniform sources and finds the threshold values for several capacity approaching LDPC codes. A critical comparison of turbo and LDPC codes for nonuniform sources is presented in Section 4. Finally Section 5 concludes the paper.

2. ENCODING AND DECODING OF NONUNIFORM SOURCES

Given the input vector x of length n , the syndrome vector s is an encoder output and is given by

$$s = Hx$$

where H is a $m \times n$ LDPC matrix. In LDPC syndrome sequence decoding for nonuniform sources, the source distribution is incorporated while initializing the edges originating from variable nodes (or bit nodes) of the *bipartite graph* [10] as derived in (4). The side information is modeled as an output of an equivalent binary symmetric channel (BSC).

We now define the following terms

q = Crossover probability of the equivalent BSC channel modeled for side information

p_0 = Source distribution, i.e $P(X = 0)$

u = Log-likelihood ratio (LLR) value of the channel information

$$u = \log \frac{1-q}{q} \quad (2)$$

v = a-priori LLR value

$$v = \log \frac{p_0}{1-p_0} \quad (3)$$

$L_{i,0}$ = Initial LLR value of i^{th} variable node

Let $x_i, y_i \in \{0, 1\}$ be the realization of the nonuniform source X (to be compressed) and side information Y respectively. The subscript ' i ' denotes that they also belong to i^{th} variable node. The edges emanating from variable nodes are initialized with

$$\begin{aligned} L_{i,0} &= \log \frac{p(x_i=0/y_i)}{p(x_i=1/y_i)} = \log \frac{p(y_i/x_i=0)P(x_i=0)}{p(y_i/x_i=1)P(x_i=1)} \\ &= (1-2y_i) \log \frac{1-q}{q} + \log \frac{p_0}{1-p_0} \quad i=1,2,\dots,n \quad (4) \\ &= (1-2y_i)u + v \end{aligned}$$

where $q = P(X \neq Y) < 0.5$

Here, the term $\log(p_0/(1-p_0))$ in (4) comes naturally from the Bayes rule and it is zero for uniform source. The rest of decoding algorithm is same as that of [9] and omitted for the sake of brevity. The block diagram of LDPC encoding and decoding of syndrome sequence is shown in Fig. 1.

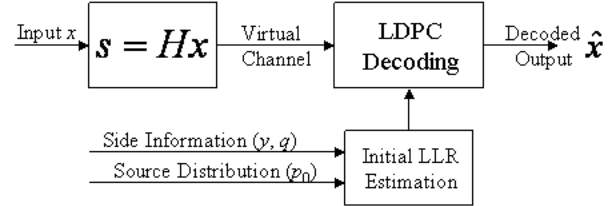


Figure 1 LDPC Encoder and Decoder with Syndrome Coding

3. DENSITY EVOLUTION OF LDPC CODES FOR NONUNIFORM SOURCES IN SLEPIAN-WOLF CODING

Density evolution [11] is a powerful tool to quickly determine the asymptotic behavior of LDPC codes for different channel and noise conditions. It evaluates the way in which the probability density function (pdf) of the variable nodes evolves with iterations for a given degree distribution [10]. The density evolution analysis starts with an initial pdf, which depends on the channel condition and a-priori information.

In this paper, we have used *discretized density evolution* [12], [13] to determine the threshold¹ values (q^*) for some best known LDPC codes for compression of nonuniform sources under DSC paradigm. The density evolution analysis helps in evaluating and comparing different LDPC codes by estimating how far they are from *optimal*.

3.1 Slepian-Wolf Bounds for Nonuniform Sources

Slepian-Wolf bound is a theoretical limit, which specifies the achievable lowest bit rate for lossless compression of a source X with another correlated source Y available at decoder as a lossless side information. In DSC paradigm, a code is said to be optimal when the bit rate approaches Slepian-Wolf bound. The density evolution thresholds are compared with the Slepian-Wolf bounds. The Slepian-Wolf bound $H(X/Y)$ is given by

$$H(X/Y) = H(X, Y) - H(Y) \quad (5)$$

assuming

$$P(X = x/Y = y) = P(Y = y/X = x)$$

we get

¹ worst case channel condition for which the error free communication is possible

$$H(X/Y) = G((1-q)p_0) + G(qp_0) \\ + G(q(1-p_0)) + G((1-q)(1-p_0))$$

where $G(\cdot)$ is defined as

$$G(x) = -x \log_2(x)$$

and

$$H(Y) = H_2((1-q)p_0 + q(1-p_0))$$

where $H_2(\cdot)$ is a binary entropy function and is given by

$$H_2(x) = -x \log_2(x) - (1-x) \log_2(1-x)$$

3.2 Discretized Density Evolution

In this section, we explore the suitability and optimality of existing LDPC codes for nonuniform sources through density evolution technique. In density evolution, the pdf of LLR messages² are being exchanged between the variable and check nodes in bipartite graph instead of LLR messages itself as in the decoding algorithm. In practice, the numerical computation of density evolution is carried out using a quantized pdf of LLR messages. This quantized version of density evolution is known as *discretized density evolution*. This discretization maintains the symmetry of the messages around zero.

The discretization function $Q(w)$ of LLR message w is defined as follows[12], [13].

$$Q(w) = \begin{cases} \left\lfloor \frac{w}{\Delta} + \frac{1}{2} \right\rfloor \cdot \Delta, & \text{if } w \geq \frac{\Delta}{2} \\ \left\lceil \frac{w}{\Delta} - \frac{1}{2} \right\rceil \cdot \Delta, & \text{if } w \leq -\frac{\Delta}{2} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where Δ is the quantization interval and $\lfloor \cdot \rfloor$, $\lceil \cdot \rceil$ are 'floor' and 'ceil' operators respectively. According to [13] the discretized density evolution is given by

$$p_u^{(l+1)} = \rho \left(\left[\lambda \left(\left[p_{u_0}^l \right] \right) \right] \right) \quad (7)$$

where p_{u_0} is the quantized initial probability mass function (pmf),

l is the iteration number, p_u^l is the quantized pmf at l^{th} iteration and $p_u^{(0)}$ has all mass at 0. Further, $\lambda(\cdot)$ and $\rho(\cdot)$ defines the degree distribution of LDPC codes. The detailed discussion of discretized density evolution algorithm is described in [12], [13].

3.3 Initial probability distribution function

While initializing the variable nodes for LDPC decoding the information about the source distribution (a-priori LLR value) is added to channel information (see (4)). Hence, in density evolution the probability density function of the channel

information is convolved with a-priori probability density function.

In discretized density evolution, the source distribution p_0 is fixed and the threshold q^* for a given degree distribution is determined. Let $P_q(x)$ be the probability density function of a BSC channel with crossover probability q and is given by [11]

$$P_q(x) = q\delta(x+u) + (1-q)\delta(x-u)$$

Similarly the pdf of source distribution $P_{p_0}(x)$ is given by

$$P_{p_0}(x) = (1-p_0)\delta(x+v) + p_0\delta(x-v)$$

and the initial pdf, $P_0(x)$, for LDPC decoder is given by

$$P_0(x) = P_q(x) \otimes P_{p_0}(x) \\ = q(1-p_0)\delta(x+u+v) \\ + qp_0\delta(x+u-v) \\ + (1-q)(1-p_0)\delta(x-u+v) \\ + (1-q)p_0\delta(x-u-v)$$

where \otimes denotes discrete convolution. The discretized version of this $P_0(x)$ is used as a initial probability mass function in (7), i.e., $p_{u_0} = Q(P_0(x))$

3.4 Specification of LDPC codes analyzed

Regular Codes

The details of the regular LDPC codes [11] analyzed in this paper are tabulated in Table 1. The code parameters and threshold values are also tabulated in Table 1. The symbols d_c and d_v indicate the column and row weights of the regular LDPC matrix, $\epsilon^*(\text{BF})$ is the threshold when decoded with bit-flipping algorithm and $\epsilon^*(\text{BP})$ is the threshold for belief propagation decoding for BSC channel. Further, σ^* indicates the threshold of the code under AWGN channel. The channel capacities for BSC and BIAWGN (Binary Input Additive White Gaussian Noise) channels are represented by ϵ_{opt} and σ_{opt} respectively. These threshold values indicate the relative performance the codes in different channel conditions and with different decoding algorithms

Irregular Codes

The degree distribution of irregular LDPC codes [14] analyzed in this paper are given in Table 2. As the code rate is equal to $R = 1/2$, the bit rate comes out to be 0.5 bits/sample. In the last row of Table 2, σ^* indicates the threshold of the code under AWGN channel. The Gap_{awgn} is the difference between $\sigma_{\text{opt}} = 0.9787$ (for 1/2 rate codes) [14, pp:623] and σ^* .

² LLR values and LLR messages are used interchangeably

Table 1: Regular Codes

Code ID	d_v	d_c	Code Rate	Bit Rate (bits/sample)	ε^* (BF)	ε^* (BP)	ε_{opt}	σ^*	σ_{opt}
code_1	3	5	0.4	0.6	0.061	0.113	0.146	1.00	1.148
code_2	4	6	0.333	0.667	0.066	0.116	0.174	1.01	1.295
code_3	3	4	0.25	0.75	0.106	0.167	0.215	1.26	1.549

Table 2: Degree distribution of some capacity approaching irregular LDPC codes

Degree Distribution	code_4 [14]	code_5 [14]	code_6 [14]
λ_2	0.38354	0.23802	0.17120
λ_3	0.04237	0.20997	0.21053
λ_4	0.57409	0.03492	0.00273
λ_5		0.12015	
λ_7		0.01587	0.00009
λ_8			0.15269
λ_9			0.09227
λ_{10}			0.02802
λ_{14}		0.0048	
λ_{15}		0.37627	0.01206
λ_{30}			0.07212
λ_{50}			0.25830
ρ_5	0.24123		
ρ_6	0.75877		
ρ_8		0.98013	
ρ_9		0.01987	0.33620
ρ_{10}			0.08883
ρ_{11}			0.57497
Performance measures [14]	$\sigma^* = 0.9114$ $Gap_{awgn} = 0.0673$	$\sigma^* = 0.9622$ $Gap_{awgn} = 0.0165$	$\sigma^* = 0.9718$ $Gap_{awgn} = 0.0069$

3.5 Details of Discretized Density Evolution and Threshold Values (q^*)

The discretized density evolution was carried out with 10 bit precision and LLR bounds taken from -25 to +25. With this LLR bounds the LLR values which fall below -25 are accounted at -25 and similarly the LLR values above +25 are accounted at +25 for calculation of evolving probability density functions of variable nodes. A probability of error less than 10^{-6} is taken as lossless and threshold values are estimated. We found that a 10 bit precision ($\Delta = 0.04883$) for discretization of LLR messages is a good trade-off between computational complexity and accuracy of the threshold values obtained. The threshold values for the regular and irregular LDPC codes given in Table 1 and 2 respectively are determined with these parameters and are tabulated in Table 3 and 4. The 'Gap' in the tables indicates the difference between the bit rate of the compressed data and the Slepian-Wolf bound, $H(X/Y)$, given by (5) with $q = q^*$.

In some cases, even when $q = 0.5$, the probability of error falls below 10^{-6} . When $q = 0.5$ the side information is completely decorrelated with the source being compressed and $u = 0$ from (2).

Hence side information has no role to play in decoding according to (4). This condition is shown as '---' in tables.

3.6 Discussion on LDPC Thresholds and Comparison with Turbo Codes

There are some striking differences between the performance of LDPC and turbo codes with respect to nonuniform sources. We have reported the turbo coding results from [5] in Table 5 for comparison purpose. This table shows that the *attainable q* (equivalent to q^* in LDPC codes) for different source distributions for (31, 23) turbo code with bit rate fixed at 0.667 bits/sample. The Gap A, B and C refers to gap between Slepian-Wolf bound and bit rate of (i) 'Berrou's code + fixed length bin-indices', (ii) 'optimal code + fixed length bin-indices' and (iii) 'optimal code + variable length bin-indices' respectively. It is clear from the Table 5 that the *attainable q* is almost constant with source distribution with both Berrou's code [8] and optimized turbo code (designed by the authors of [5] for non-uniform sources). It implies that the turbo code performance is invariant to the source distribution. Hence, variable bin indices are necessary

Table 3: Threshold values for regular codes

	code_1			code_2			code_3		
p_0	q^*	H(X/Y)	Gap to SW bound	q^*	H(X/Y)	Gap to SW bound	q^*	H(X/Y)	Gap to SW bound
0.70	0.135	0.5147	0.0853	0.138	0.5216	0.1454	0.204	0.6520	0.0980
0.75	0.153	0.5173	0.0827	0.155	0.5211	0.1459	0.238	0.6531	0.0969
0.80	0.185	0.5185	0.0815	0.187	0.5214	0.1456	0.310	0.6529	0.0971
0.85	0.257	0.5172	0.0828	0.263	0.5220	0.1450	----	----	----
0.90	----	----	----	----	----	----	----	----	----

Table 4: Threshold values for irregular codes

	code_4			code_5			code_6		
p_0	q^*	H(X/Y)	Gap to SW bound	q^*	H(X/Y)	Gap to SW bound	q^*	H(X/Y)	Gap to SW bound
0.70	0.100	0.4255	0.0745	0.119	0.4759	0.0241	0.122	0.4834	0.0166
0.75	0.113	0.4311	0.0689	0.135	0.4806	0.0194	0.138	0.4870	0.0130
0.80	0.136	0.4380	0.0620	0.163	0.4848	0.0152	0.166	0.4896	0.0104
0.85	0.180	0.4400	0.0600	0.220	0.4839	0.0161	0.225	0.4888	0.0112
0.90	0.300	0.4255	0.0745	----	----	----	----	----	----

to exploit the redundancy left in the turbo binning scheme as proposed in [5]. In case of LDPC codes, it is apparent from the Tables 3 and 4, that the q^* varies with the source distribution. More the non-uniformity of the source, more is the simulation threshold. The larger value of q^* indicates that the code can tolerate weakly correlated side information. A significant thing to be noted (unlike, in case of turbo codes), the entropy H(X/Y) of nonuniform source with side information is almost constant when the LDPC codes are used optimally i.e., at threshold values. Since we have used a 10bit quantization, the H(X/Y) appears to be constant only upto two digits after decimal point. We conjecture that, if infinity precision is used, the H(X/Y) will be a constant.

Conjecture 1: At threshold values of LDPC codes, the conditional entropy H(X/Y) is constant with source distribution of X, when X is coded with LDPC syndrome sequence with the lossless side information Y.

One nice application of this property is that, this result can be used to find the thresholds (q^*) at different source distributions. It is sufficient to perform only one density evolution analysis at some source distribution to find the corresponding q^* and find the value of H(X/Y).

Example 1: Assume that a threshold value q_0^* is found through density evolution at some source distribution, $p_{x0} = P(X = 0)$ with infinity precision. The conditional entropy H(X/Y) can be expanded as

$$H(X/Y) = H(p_{x0}) + H(q_0^*) - H(p_{y0}) \quad (8)$$

where

$$p_{y0} = P(Y = 0) = p_{x0}(1 - q_0^*) + (1 - p_{x0})q_0^*$$

Now, we would like to find the threshold value (q_1^*) at some other source distribution say p_{x1} . The following equation can be solved for q_1^* using Newton-Raphson method.

$$H(p_{x1}) + H(q_1^*) - H(p_{y1}) = H(X/Y) \quad (9)$$

where

$$p_{y1} = p_{x1}(1 - q_1^*) + (1 - p_{x1})q_1^*$$

and H(X/Y) is known from (8).

From the above discussion, it is clear that the LDPC codes will adapt to source distribution and an optimally used LDPC code (i.e., at threshold values) can achieve the Slepian-Wolf bound without any further entropy coding. Now it is curious to know, why LDPC codes respond to source distribution, while turbo codes are immune. This aspect is discussed in Section 4.

4. DISCUSSION ON TURBO AND LDPC CODES FOR NONUNIFORM SOURCES

Generally in distributed source coding, turbo codes are used in syndrome former (SF) and inverse syndrome former (ISF) setup to achieve maximum compression and are compatible with syndrome sequence coding of LDPC codes for comparison purpose. The SF-ISF formation is shown in Fig. 2, where the side information and an arbitrary sequence from the coset (represented by the syndrome former) are added. The added sequence (\hat{c}) forms a corrupted codeword and enters into the conventional turbo decoder for error correction decoding. Assuming that the turbo decoder corrects all the errors, we add the corrected codeword to the arbitrary sequence from the ISF to recover the original sequence (i.e., x). In turbo decoder the constituent decoders exchange the extrinsic information back and forth. The extrinsic information of one decoder is used as a priori information to the other constituent decoder. Let $L_{i,ext}^{(l)}(u_j)$ be the extrinsic information of j^{th} bit from i^{th} decoder at l^{th} iteration. Similarly, let $L_{i,app}^{(l)}(u_j)$ be a priori information to j^{th} bit in i^{th} decoder ($i \in \{1, 2\}$).

$$L_{1,app}^{(l)}(u_j) = L_{2,ext}^{(l-1)}(u_j), L_{2,app}^{(l)}(u_j) = L_{1,ext}^{(l-1)}(u_j) \quad (10)$$

Table 5: Turbo Code Performance Results from [5]

The (31, 23) Turbo Code with bit rate fixed at 2/3 bits/sample for different source distribution p_0							
	Berrou's Code + Fixed Length Bin-Indexes			Optimal Code + Variable Length Bin-Indexes			
p_0	attainable q	H(X/Y)	Gap A	attainable q	H(X/Y)	Gap B	Gap C
0.7	0.139	0.5239	0.1427	0.143	0.5330	0.1337	0.1213
0.8	0.139	0.4435	0.2232	0.143	0.4507	0.2159	0.1522
0.9	0.136	0.3035	0.3632	0.141	0.3090	0.3574	0.1444

For nonuniform source distribution, a modification was suggested in [5] to the above equations as follows.

$$L_{1,app}^{(l)}(u_j) = L_{2,ext}^{(l-1)}(u_j) + \log \frac{p_0}{1-p_0} \quad (11)$$

$$L_{2,app}^{(l)}(u_j) = L_{1,ext}^{(l-1)}(u_j) + \log \frac{p_0}{1-p_0} \quad (12)$$

where p_0 is the source distribution and the term $\log(p_0/(1-p_0))$ is included in every iteration to account for nonuniform source distribution. This inclusion is questionable, and we emphasis more on this point in the next paragraph.

The arbitrary sequence from ISF, a member of the coset generally do not to have the same source distribution as input sequence x . Further, the input to the turbo decoder is XOR of ISF sequence and side information. Hence, the source distribution of the input to the system and the input to the turbo decoder are generally different. For example, assume that ISF gives an arbitrary sequence equal to that of input sequence. Also assume that the side information is also equal to the input. In this scenario, irrespective of input sequence, all zero sequence will be the input to turbo decoder. This simple example shows that it is not possible to estimate the source distribution of input to the turbo decoder in SF-ISF setup. But in [5] authors, suggested to include the information about the source distribution in a *a priori term* for each constituent decoder as per (11) and (12). As already emphasized, the source distribution doesn't reflect in the input to the turbo decoder and hence the inclusion of the term, $\log(p_0/(1-p_0))$, in iterative decoding is not justified. Because of this reason the turbo coding performance is invariant to source distribution and has no effect on 'attainable q '. This is readily evident from the Table 5.

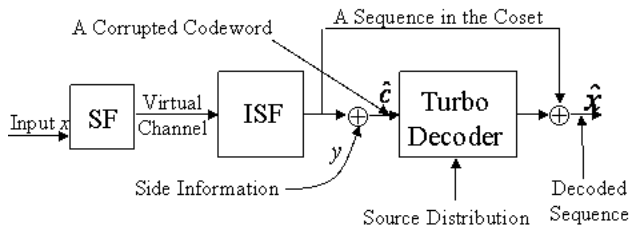


Figure 2 Turbo Encoder and Decoder with SF-ISF

In case of LDPC codes, the encoding and decoding in error correction setup and syndrome coding setup are very much similar. In DSC, generally a non-zero syndrome is sent to decoder, whereas in error correction the syndrome is always zero. In LDPC

decoding for error correction, each bit in the codeword is decoded and information bits are extracted from the decoded codeword. The LDPC decoder does not distinguish between information bits and parity bits, whereas the turbo decoder clearly distinguishes the parity bits from information bits. In LDPC decoding we can naturally incorporate the source distribution for each bit in the codeword as shown in Sec 2. In turbo decoding, the extrinsic information for source bits are only computed and passed to the other constituent decoder as a priori information. But in SF-ISF setup, both source and parity bit positions constitute the input. So we cannot truly incorporate the source distribution for each input bit in the SF-ISF turbo decoder.

5. CONCLUSION

In this paper, the LDPC codes are analyzed for compression of nonuniform sources under Slepian-Wolf coding using density evolution technique. Threshold values for several capacity approaching LDPC codes were determined through discretized density evolution technique at different source distributions. These threshold values show that, the performance of the capacity approaching LDPC codes is close to Slepian-Wolf bound. From Table 3 and 4 we find that, for a given LDPC code, the conditional entropy H(X/Y) is nearly constant with source distribution probabilities (p_0). Hence, we can analytically determine the q^* for a given p_0 without going through density evolution technique. From this observation, we may state that as the non-uniformity of the source distribution increases, the dependence on side information (i.e., larger value of q^*) decreases for extraction of the source information at decoder. The proposed method is a one step approach, in the sense that it does not require entropy coding on syndrome sequence as it is necessary for turbo binning scheme.

6. REFERENCES

- [1] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources", in IEEE Trans on Information Theory, Vol. 19, No. 4 July 1973, pp: 471-480.
- [2] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", in IEEE trans on Information Theory, Vol. 22, No. 1, Jan 1976, pp: 1-10.
- [3] S. S. Pradhan and K. Ramachandran, "Distributed source coding using syndromes (DISCUS): Design and construction", in IEEE Trans on Information Theory, Vol. 49, No. 3, March 2003, pp: 626-634.
- [4] J. Kroll and N. Phamdo, "Source-channel optimized trellis codes for bi-tonal image transmission over awgn channels" in IEEE Trans on Image Processing, July 1999, pp: 899-912.

- [5] Jing Li, Zhenyu Tu and Rick S Blum, “Slepian-Wolf Coding for Nonuniform Sources Using Turbo Codes” in Proceedings of IEEE Data Compression Conference (DCC) pp : 312-321, Snowbird, UT, March 2004.
- [6] V Toto Zarasoa, et.al., “On Distributed Arithmetic Codes and Syndrome Based Turbo Codes for Slepian-Wolf Coding of Nonuniform Sources”, in 17th European Signal Processing Conference (EUSIPCO 2009), Glasgow, Scotland, Aug 24-28, 2009.
- [7] Grangetto M, et al., “Distributed Arithmetic Coding for the Slepian-Wolf Problem”, in IEEE Trans on Signal Processing , Vol. 57, No. 6, June 2009, pp: 2245-2257.
- [8] Berrou C, Glavieux A and Thitimajshima, P, “Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes” in IEEE Proceedings of the Int. Conf. on Communications, Geneva, Switzerland, May 1993 (ICC 1993) pp. 1064-1070.
- [9] A.D.Liveris, Z. Xiong and C.N. Georghiadis, “Compression of Binary Sources With Side Information at the Decoder Using LDPC Codes”, in IEEE Communications Letters, Vol. 6, No. 10, Oct 2002, pp: 440-442.
- [10] William E Ryan, “An Introduction to LDPC Codes”, available online www.ece.arizona.edu/~ryan/publications/ldpcchap.pdf.
- [11] T. J. Richardson and R. L. Urbanke, “The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding”, in IEEE Trans on Information Theory, Vol. 47, No. 2 Feb 2001, pp: 599-618.
- [12] S. Y. Chung, “On the Construction of Some Capacity-Approaching Coding Schemes” PhD Thesis MIT, Sept 2000.
- [13] S. Y. Chung, G. D. Forney, T. J. Richardson and R. L. Urbanke, “On Design of Low-Density Parity-Check Codes within 0.0045 dB from Shannon Limit”, in IEEE Comm. Letters, Vol, 5, No. 2, Feb 2001, pp: 58-60.
- [14] T. J. Richardson, M. A. Shokrollahi and R. L. Urbanke, “Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes”, in IEEE Trans on Information Theory, Vol. 47, No.2 Feb 2001, pp:619-637.