Rate-Adaptive Codes for Distributed Source Coding *

David Varodayan†, Anne Aaron and Bernd Girod

Information Systems Lab., Dept. of Electrical Engineering
Stanford University, Stanford, CA 94305, USA

Abstract

Source coding with correlated decoder side information is considered. We impose the practical constraint that the encoder be unaware of even the statistical dependencies between source and side information. Two classes of rate-adaptive distributed source codes, both based on Low-Density Parity-Check (LDPC) codes, are developed and their design is studied. Specific realizations are shown to be better than alternatives of linear encoding and decoding complexity. The proposed rate-adaptive LDPC Accumulate (LDPCA) codes and Sum LDPC Accumulate (SLDPCA) codes (of length 6336 bits) perform within 10% and 5% of the Slepian-Wolf bound in the moderate and high rate regimes, respectively.

Keywords: distributed source coding, Slepian-Wolf theorem, side information, LDPC codes, iterative decoding algorithms

1. Introduction

Asymmetric distributed source coding is depicted in Fig. 1. A finite-alphabet source X is to be transmitted without loss using the least average number of bits. Statistically dependent side information Y (not necessarily discrete) is available at the decoder only. The encoder must therefore compress X in the absence of Y, whereas the decoder uses Y to aid the recovery of X. Slepian and Wolf proved in 1973 that lossless compression is achievable at rates $R \geq H(X|Y)$, the conditional entropy of X given Y, for X and Y discrete [1]. Observe that this rate bound is the same as if Y were known to the encoder as well as the decoder. Wyner and Ziv extended this result to the cases of lossy compression [2] and non-discrete X and Y [3].

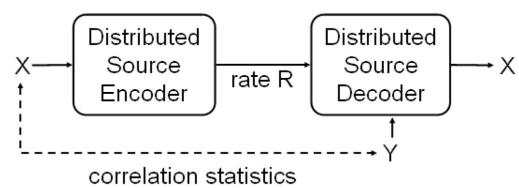

Fig. 1. The asymmetric distributed source coding scenario

The application of channel codes to source coding problems was proposed by Blizard [4] in 1969 and Hellman [5] in 1975. Slepian and Wolf [1] and Wyner [6] noted the relationship between channel coding and source coding with side information. Pradhan and Ramchandran revived the approach with their DISCUS framework [7]. The distributed source encoder compresses X into its syndrome S with respect to a channel code $\mathcal{C}$. Upon receipt of the syndrome, the distributed source decoder can narrow down the possible values of X to the coset represented by S in $\mathcal{C}$. It then disambiguates X from among these coset el-

*This work was supported by NSF under Grant No. CCR-0310376. It has been presented in part at the Asilomar Conference on Signals, Systems, and Computers 2005, in Pacific Grove, CA.
†Corresponding author. Tel.: +1-650-723-3476. E-mail address: varodayan@stanford.edu (D. Varodayan).

ements as the most likely one given the correlated side information Y. The impressive potential of this approach has been demonstrated by various implementations of the system with turbo codes [8] [9] [10] and Low-Density Parity-Check (LDPC) codes [11]. In these schemes, achieving compression close to the Slepian-Wolf bound depends on choosing an appropriate channel code; the dependencies between X and Y play the role of the channel statistics in this context. If the statistics of this 'dependency channel' are known at both encoder and decoder, they can agree on a good code and a rate close to the Slepian-Wolf limit can be used.

For many practical applications, the statistical dependency between X and Y may not be known at the encoder. Low complexity video coding via distributed source coding, for example, treats a frame of video as the source X and its prediction at the decoder as the side information Y [12]. Since video data are highly non-ergodic, the achievable compression ratio varies and cannot be foretold by the encoder. In this situation, a rate-adaptive scheme with feedback is an attractive solution. The encoder transmits a short syndrome based on an aggressive code and the decoder attempts decoding. In the event that decoding is successful, the decoder signals this fact to the encoder, which then continues with the next block of source data. However, if decoding fails, the encoder augments the short syndrome with additional transmitted bits, creating a longer syndrome based on a less aggressive code. The process loops until the syndrome is sufficient for successful decoding. Obviously, this approach is viable only if a feedback channel is available and the round-trip time is not too long.

Punctured turbo codes [13] were used to implement rate-adaptive source coding with and without decoder side information in [12] and [14], respectively. The latter addressed the design of the codes and puncturing via EXIT chart analysis [15]. Punctured LDPC codes [16] [17] [18], though they may be applied to this problem, perform poorly after even moderate puncturing. Recently, Chen et al. presented a better rate-adaptive distributed source coding architecture with decoding complexity of $O(n \log n)$ [19], where n is the blocklength of the code.

The primary contribution of this paper is the development and design of LDPC-based rate-adaptive distributed source codes that have better performance than alternative codes of linear complexity encoding and decoding. In Sec. 2 and 3, we introduce rate-adaptive LDPC Accumulate (LDPCA) codes and Sum LDPC Accumulate (SLDPCA) codes, respectively. Sec. 4 discusses properties of these codes and strategies for their design. In Sec. 5, we compare the performance of the two proposed classes of rate-adaptive distributed source codes with the performance of the turbo-coded system of [12] over various code lengths and source and conditional statistics.

2. LDPC Accumulate (LDPCA) Codes

As noted in Sec. 1, LDPC codes (in syndrome code form) have been used effectively in fixed-rate distributed source coding [11]. A naïve way to use such a code as part of a rate-adaptive scheme would be to transmit the syndrome bits in stages and allow decoding after receipt of each increment of the syndrome. However, the performance of the high compression codes so derived is very poor because their graphs contain unconnected or singly-connected source nodes; these structural features impede the transfer of information via the LDPC iterative decoding algorithm [20] [21]. Instead, we now present a method for constructing LDPC-based rate-adaptive codes for distributed source coding, whose performance does not degrade at high compression ratios.

The LDPCA encoder consists of an LDPC syndrome-former concatenated with an accumulator. An example is shown in Fig. 2. The source bits $(x_1, \ldots, x_8)$ are first summed modulo 2 at the syndrome nodes according to the LDPC graph structure, yielding syndrome bits $(s_1, \ldots, s_8)$. These syndrome bits are in turn accumulated modulo 2, producing the accumulated syndrome $(a_1, \ldots, a_8)$. The encoder buffers the accumulated syndrome and transmits it incrementally to the decoder. This encoder structure can be recast straightforwardly as the encoder for an extended Irregular Repeat Accumulate (eIRA) channel code [22]. The eIRA interpretation, however, treats the accumulated syndrome nodes as degree 2 variable nodes, inducing a degree distribution different to that of the underlying LDPC code. Section 4 demonstrates that only this underlying degree distribution is invariant under the rate-adaptive operation of LDPCA codes. Hence, we avoid conflating the concepts of LDPCA and eIRA codes.

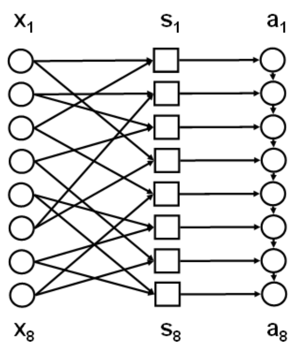

Fig. 2. The LDPCA encoder

The LDPCA decoder handles rate-adaptivity by modifying its decoding graph each time it receives an additional increment of the accumulated syndrome. First assume, for the sake of argument, that the entire accumulated syndrome $(a_1, \ldots, a_8)$ has been received. Then taking the consecutive differences modulo 2 of these values yields the syndrome $(s_1, \ldots, s_8)$. The syndrome-adjusted LDPC iterative decoding method of [11] can be applied on the same graph (shown in Fig. 3a) that was used for encoding $(s_1, \ldots, s_8)$ from $(x_1, \ldots, x_8)$. For decoding, the source nodes are seeded with conditional probability distributions of the source bits given the side information, namely $\Pr\{X_1|Y\}, \ldots, \Pr\{X_8|Y\}$. Then messages are passed back and forth between the source nodes and the syndrome nodes (according to the equations in [11]) until the estimates of the source bits converge. The correctness of the recovered source values can be tested with respect to the syndrome bits, with very small chance of a false positive. When the number of received bits equals the number of source bits, as in this case, the performance achieved by transmitting $(a_1, \ldots, a_8)$ is no different to that had by transmitting $(s_1, \ldots, s_8)$ since the resulting decoding graphs are identical.

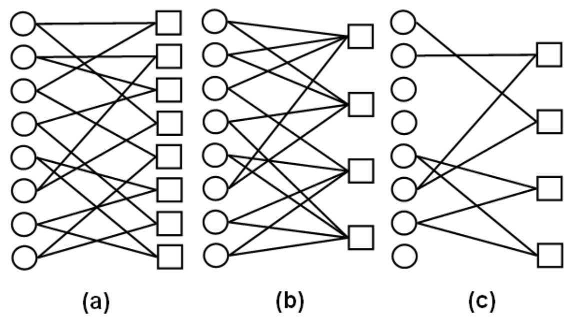

Fig. 3. Decoding graphs if the encoder transmits (a) the entire accumulated syndrome, (b) the even-indexed accumulated syndrome bits, (c) the even-indexed syndrome bits.

The modification of decoding graph structure manifests at higher compression ratios. Consider, for instance, a compression ratio of 2. In our example, this corresponds to the transmission of only the even-indexed subset of the accumulated syndrome (a_2, a_4, a_6, a_8). The consecutive difference modulo 2 operation at the decoder then produces $(s_1 + s_2, s_3 + s_4, s_5 + s_6, s_7 + s_8)$. Fig. 3b shows the graph which would have encoded $(s_1 + s_2, s_3 + s_4, s_5 + s_6, s_7 + s_8)$ from $(x_1, \ldots, x_8)$. This graph maintains the degree of all source nodes compared to Fig. 3a. Therefore, it can be used for effective iterative decoding with source bit seeding $\Pr\{X_1|Y\}, \ldots, \Pr\{X_8|Y\}$. Upon completion of decoding, the recovered source can be tested against the syndrome to verify correctness. For comparison, if the syndrome subset (s_2, s_4, s_6, s_8) were transmitted instead of the accumulated syndrome subset, the decoding graph (shown in Fig. 3c) would be severely degraded and unsuitable for iterative decoding.

Finally, note that the encoding and decoding complexity of these LDPCA codes is linear in the number of edges, which is invariant under the proposed construction. Moreover, the number of edges is linear in the length of the code for a fixed degree distribution. Therefore, the complexity of encoding and decoding is $O(n)$, where n is the blocklength of the code in bits.

3. Sum LDPC Accumulate (SLDPCA) Codes

Serially Concatenated Accumulate codes have been proposed for the rate-adaptive distributed source coding problem [19]. For this class of codes, the encoder is the concatenation of an inverse accumulator with one or more rate-adaptive base codes. The base codes considered in [19] are simple product codes and extended Hamming codes, yielding overall Product Accumulate codes [23] and extended Hamming Accumulate codes [24]. Both of these systems incur decoding complexity of up to $O(n \log n)$ since that is the soft decoding complexity for the base codes. In this section, we consider using LDPCA codes of Sec. 2 as base codes to create rate-adaptive SLDPCA codes, which are linear in encoding and decoding complexity with respect to their code lengths.

The SLDPCA encoder is the concatenation of a consecutive summer (i.e. inverse accumulator) with an LDPCA encoder; an example is depicted in Fig. 4. Here, the source bits $(x_1, \dots, x_8)$ are consecutively summed into intermediate bits $(i_1, \dots, i_8)$, which are then coded in the same fashion as in Fig. 2 to produce the accumulated syndrome $(a_1, \dots, a_8)$. As with the previous rate-adaptive scheme, the encoder buffers the accumulated syndrome and transmits it incrementally to the decoder.

Although the SLDPCA decoder employs a different decoding algorithm, its rate-adaptive functionality is the same as that of the decoder in Sec. 2. That is, when the decoder receives each increment of accumulated syndrome bits from the encoder, it modifies the LDPC portion of its decoding graph to reflect this information.

To understand the decoding algorithm, first observe the following property of the encoder in Fig. 4: the source bits $(x_1, \dots, x_8)$ are the accumulated sum of the intermediate bits $(i_1, \dots, i_8)$ modulo 2. In other words, $(x_1, \dots, x_8)$ are the output of a very simple IIR filter with input $(i_1, \dots, i_8)$. This means that the decoder can

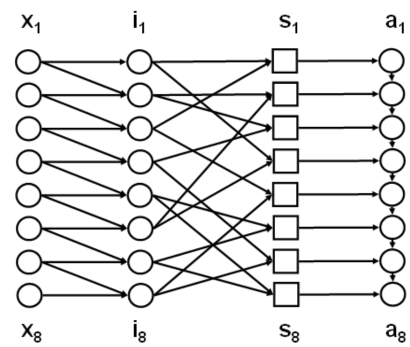

Fig. 4. The SLDPCA encoder

employ the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [25] to obtain soft information about $(i_1, \dots, i_8)$ when the source nodes are seeded with the conditional probability distributions $\Pr\{X_1|Y\}, \dots, \Pr\{X_8|Y\}$. Meanwhile, a single iteration of the syndrome-adjusted LDPC decoding algorithm of [11] provides different soft information about $(i_1, \dots, i_8)$ originating from the received accumulated syndrome bits. The soft information interchange at the intermediate nodes works as follows: each incoming message to an intermediate node is answered by the net information of all other incoming message to that node. Simultaneous BCJR and LDPC decoding iterations follow. Decoding continues in this way until the estimates of the intermediate bits converge. Finally, the source bits can be recovered from the decoded intermediate bits by accumulation modulo 2. Just like LDPCA codes, the correctness of the recovered source values can be tested with respect to the syndrome bits, with very small chance of a false positive.

The decoding complexities of the BCJR and LDPC iterations are linear in the code length and the number of LDPC edges, respectively. Once again, for a rate-adaptive set of codes with fixed LDPC degree distribution on the intermediate nodes, the number of LDPC edges is linear in the code length. Hence, the overall decoding complexity is $O(n)$; the encoding complexity of SLDPCA codes is also $O(n)$.

4. Code Design

We now consider some properties of the proposed LDPCA and SLDPCA codes and show how to leverage existing LDPC design techniques.

We begin with the observation that the decoding graph of Fig. 3b is obtained from the graph of Fig. 3a by merging adjacent syndrome nodes. The edges connected to each merged node are those that were connected to one of its constituent nodes, but not both (because double edges cancel out in modulo 2). Ensuring that no edges are lost over several merging steps requires very careful design of the lowest compression ratio code.

A simpler strategy to guarantee a constant number of edges for all decoding graphs is to begin the design with the highest compression ratio graph. Knowing the transmission order of the accumulated syndrome allows the derivation of graphs for each of the lower compression ratio codes. Each additional accumulated syndrome bit received results in the division of a syndrome node into two adjacent ones. The key to maintaining a constant number of edges across all graphs is to partition the edge set of the old syndrome node into the edge sets of the new pair. Furthermore, this approach keeps invariant the degrees of the source and intermediate nodes for the LDPCA and SLDPCA codes, respectively. Thus, the global degree distribution can be used to tune the performance of the codes [26]. In fact, degree distributions optimized for LDPC channel codes (for example, using [27]) can be applied directly to LDPCA codes due to the similarity of their decoding algorithms.

When the number of received bits equals the number of source bits, there is an additional design objective: the equations that generate the accumulated syndrome bits should be independent in the source bits. So, for the coding rate of 1, this guarantees decoding of the source via straightforward linear algebra, regardless of the quality of the side information.

Constructing the graphs from the highest compression ratio code to lowest also has implications on local graph structure. Define a stopping set A to be a set of source nodes, all of whose (syndrome node) neighbors are connected to at least two nodes of A. The stopping number of a graph, the size of its smallest stopping set, affects the performance of iterative LDPC decoding on that graph [28]; the larger the stopping number is, the more likely it is that decoding will succeed. We now show that the generation of lower compression ratio codes by the division of syndrome nodes into pairs does not decrease the stopping number of the graph. Suppose that the subset of source nodes A is not a stopping set of the higher compression ratio graph. So, there exists a syndrome node s_A that is singly-connected to A. The graph for the lower compression ratio code is obtained by dividing syndrome nodes into pairs, such that the edge sets of the pair form a partition on the edge set of the original syndrome node. If s_A is not divided in this way, it remains singly-connected to A. If s_A is divided, exactly one of the resulting pair of syndrome nodes is singly-connected to A. In both cases, A is not a stopping set of the new graph. Hence, stopping sets are not created by the syndrome division operation, so the stopping number does not decrease as lower compression ratio codes are generated from higher compression ratio ones. Moreover, various stopping set elimination heuristics [29] [30] can be applied to create the LDPC graph for the lowest compression ratio code. Therefore, this construction of LDPCA and SLDPCA codes propagates good local graph structure.

The design of LDPCA and SLDPCA codes, thus, leverages existing work on degree distribution optimization and stopping set elimination.

5. Simulation Results

In this section, the performance of various LDPCA and SLDPCA codes are compared with respect to different LDPC degree distributions and code lengths. Simulation results are also presented for different conditional statistics between source and side information and different source distributions. The LDPC subgraphs of all codes presented here were constructed by the method suggested in Sec. 4: starting with the highest compression ratio graph, the other graphs are obtained by successively dividing syndrome nodes into pairs. We assume that the decoder can detect lossless recovery of the source bits perfectly. The simulations also reflect the fact that decoding is always successful if the number of received accumulated syndrome bits equals the source length, as long as the received bits are generated by independent functions of the source bits. The rate points plotted are the average of 75 trials each.

Fig. 5 compares three LDPCA code systems of differing degree distributions with rate-adaptive turbo codes, all of source length 6336 bits, with i.i.d. binary symmetric (BSC) statistics between X and Y. The turbo codes, whose encoder is specified in [13], are those used in [12]. The regular LDPCA codes have a degree distribution given by ($\delta_3 = 1$), where δ_r is the proportion of nodes of degree r. One set of irregular LDPCA codes has a degree distribution of ($\delta_2 = 0.3, \delta_3 = 0.4, \delta_4 = 0.3$). The other irregular LDPCA codes shown have the following degree distribution selected from [27]: ($\delta_2 = 0.316, \delta_3 = 0.415, \delta_7 = 0.128, \delta_8 = 0.069, \delta_{19} = 0.020, \delta_{21} = 0.052$). For comparison, we demonstrate the extremely poor performance of the underlying LDPC codes under naïve incremental transmission of syndromes. We also plot the performance of the irregular LDPC code of fixed rate 0.5 and length 10000 bits, presented in [11]. The Slepian-Wolf bound indicates the performance of an ideal distributed source code; namely, rate equal to $H(X|Y)$.

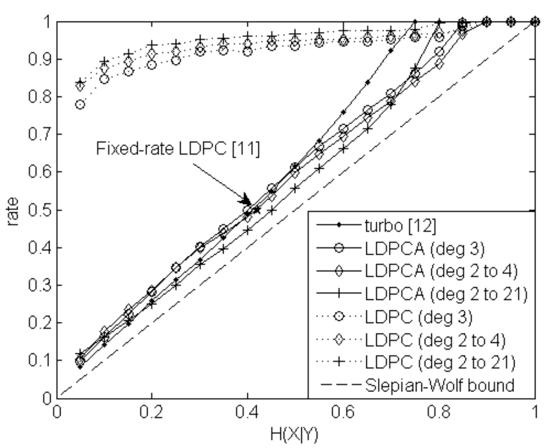

Fig. 5. Performance of regular and irregular LDPCA codes of length 6336 bits over i.i.d. BSC statistics

Fig. 6 compares two SLDPCA code systems of differing degree distributions with rate-adaptive turbo codes, all of source length 6336 bits, with i.i.d. BSC statistics between X and Y. The turbo codes are identical to those in Fig. 5. The regular SLDPCA codes have an LDPC degree distribution over intermediate nodes given by ($\delta_2 = 1$), while the intermediate nodes of the irregular SLDPCA codes have an LDPC degree distribution of ($\delta_1 = 0.3, \delta_2 = 0.4, \delta_3 = 0.3$). Figs. 5 and 6 indicate that LDPCA and SLDPCA codes are superior to turbo codes over a large range of rates. Moreover, some irregular codes can outperform their regular counterparts for the entire range of rates. We demonstrate also that irregular LDPCA codes can perform within 10% of the Slepian-Wolf bound at moderate rate, while irregular SLDPCA codes can operate within 5% of the bound at high rate.

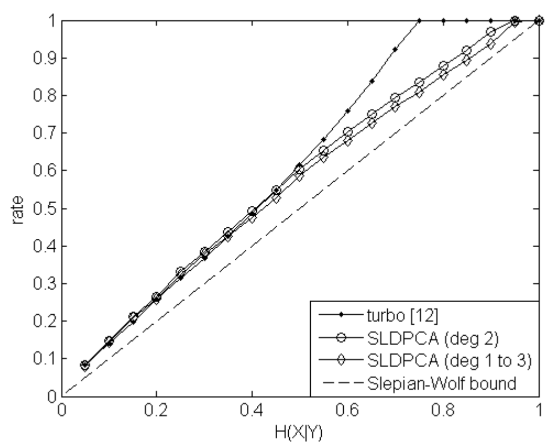

Fig. 6. Performance of regular and irregular SLDPCA codes of length 6336 bits over i.i.d. BSC statistics

The effect of varying the length of the codes is demonstrated in Fig. 7, which compares codes of length 396 and 6336 for BSC statistics between X and Y. The turbo codes are once again from [13]. Both sets of LDPCA codes have degree distribution of ($\delta_2 = 0.3, \delta_3 = 0.4, \delta_4 = 0.3$), and both sets of SLDPCA codes have degree distribution of ($\delta_1 = 0.3, \delta_2 = 0.4, \delta_3 = 0.3$). The plot indicates that reducing the length of the code degrades compression performance slightly, if at all. The discrepancy is not as large as for fixed-rate distributed source codes [11] because rate-adaptive codes are opportunistic. Over several trials, even though the maximum rate required by the short codes exceeds the maximum rate for the long ones, the minimum rate for the short ones is also less than that for the long codes.

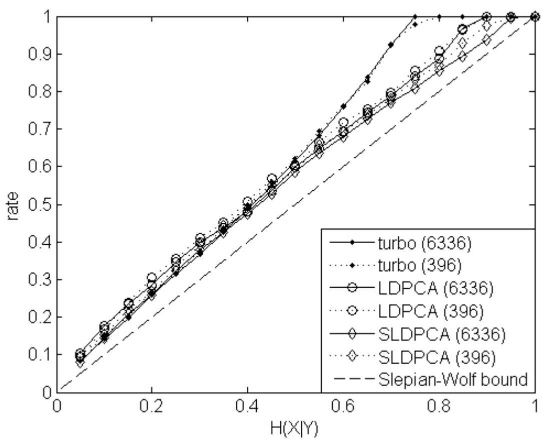

Fig. 7. Performance of rate-adaptive codes of lengths 396 and 6336 bits over i.i.d. BSC statistics

Fig. 8 investigates the application of LDPCA and SLDPCA codes to different conditional statistics between X and Y. The two models considered are i.i.d. BSC and i.i.d. Z (in which ones in X may be flipped into zeros in Y, but zeros in X cannot be flipped to ones in Y). The turbo, LDPCA and SLDPCA codes are the same length 6336 as used in Fig. 7.

Fig. 9 compares the performance of LDPCA and SLDPCA codes for BSC statistics between X and Y, when the source X is biased and unbiased. For the biased scenarios, X is modeled to be 90% zeros and 10% ones. In this plot, the turbo, LDPCA and SLDPCA codes are the same length 6336 ones as those in Fig. 7. Note that, when the source is so biased, the conditional entropy of the side information does not exceed 0.5 bits. Figs. 8 and 9 show that the performance of LDPCA and SLDPCA codes are not degraded by asymmetries in conditional statistics and source distribution.

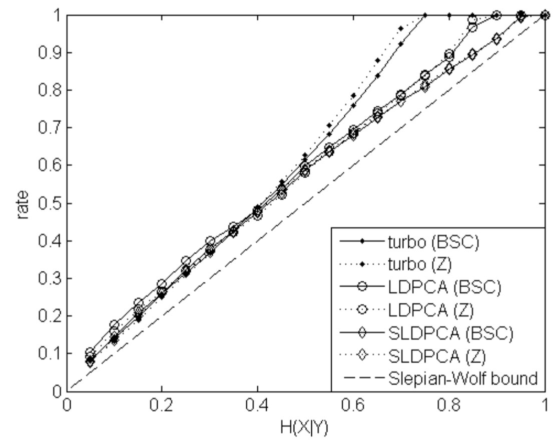

Fig. 8. Performance of rate-adaptive codes of length 6336 bits over i.i.d. Z and i.i.d. BSC statistics

6. Conclusion

This paper presents rate-adaptive LDPCA and SLDPCA codes for the case of asymmetric distributed coding in which the encoder is not aware of the joint statistics between source and side information. Our constructions guarantee the performance of the codes at all compression ratios by fixing the LDPC degree distribution across them. In addition, good local structure (with respect to stopping sets) is propagated through the graphs from low compression ratios to high.

The proposed rate-adaptive codes have been demonstrated to be superior to linear encoding and decoding complexity alternatives for asymmetric distributed source coding. LDPCA and SLDPCA codes (of length 6336 bits) are able to perform within 10% and 5% of the Slepian-Wolf bound in the moderate and high rate regimes, respectively. We have also shown that the performance of the codes diminishes only slightly when the code length is reduced, and is not degraded by asymmetries in conditional statistics and source distribution.

References

[1] D. Slepian, J. K. Wolf, Noiseless coding of correlated information sources, IEEE Trans. Inform. Theory 19 (4) (1973) 471–480.

[2] A. D. Wyner, J. Ziv, The rate-distortion function for

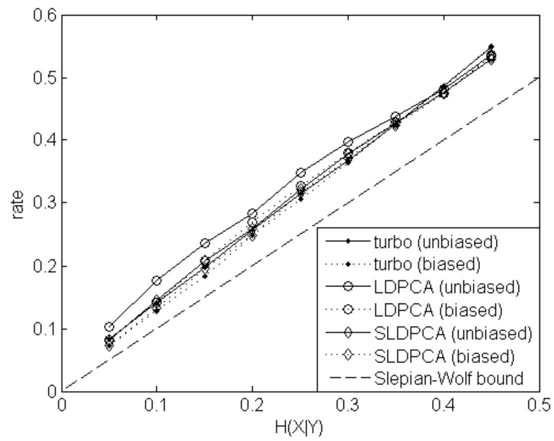

Fig. 9. Performance of rate-adaptive codes of length 6336 bits over i.i.d. BSC statistics with biased and unbiased sources

source coding with side information at the decoder, IEEE Trans. Inform. Theory 22 (1) (1976) 1–10.

[3] A. D. Wyner, The rate-distortion function for source coding with side information at the decoder-II: general sources, Inf. Control 38 (1) (1978) 60–80.

[4] R. B. Blizard, Convolutional coding for data compression, Martin Marietta Corp., Denver Div., Rep. R-69-17 (1969) (cited by [5]).

[5] M. E. Hellman, Convolutional source encoding, IEEE Trans. Inform. Theory 21 (6) (1975) 651–656.

[6] A. D. Wyner, Recent results in the Shannon theory, IEEE Trans. Inform. Theory 20 (1) (1974) 2–10.

[7] S. S. Pradhan, K. Ramchandran, Distributed source coding using syndromes (DISCUS): design and construction, IEEE Trans. Inform. Theory 49 (3) (2003) 626–643.

[8] A. Aaron, B. Girod, Compression with side information using turbo codes, in: Proc. IEEE Data Compression Conf., Snowbird, UT, 2002.

[9] J. García-Frías, Compression of correlated binary sources using turbo codes, IEEE Commun. Lett. 5 (10) (2001) 417–419.

[10] J. Bajcsy, P. Mitran, Coding for the Slepian-Wolf problem with turbo codes, in: Proc. IEEE Global Communications Conference, San Antonio, TX, 2001.

[11] A. Liveris, Z. Xiong, C. Georghiades, Compression of binary sources with side information at the decoder using LDPC codes, IEEE Commun. Lett. 6 (10) (2002) 440–442.

[12] A. Aaron, S. Rane, E. Setton, B. Girod, Transform-domain Wyner-Ziv codec for video, in: SPIE Visual Communications and Image Processing Conf., San Jose, CA, 2004.

[13] D. N. Rowitch, L. B. Milstein, On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes, IEEE Trans. Commun. 48 (6) (2000) 948–959.

[14] J. Hagenauer, J. Barros, A. Schaeffer, Lossless turbo source coding with decremental redundancy, in: Proc. 5th ITG Conf. on Source and Channel Coding, Erlangen, Germany, 2004.

[15] S. ten Brink, Designing iterative decoding schemes with the extrinsic information transfer chart, AEÜ Int. J. Electron. Commun. 54 (6) (2000) 389–398.

[16] J. Ha, S. W. McLaughlin, Optimal puncturing of irregular low-density parity-check codes, in: Proc. IEEE International Conf. on Communications, Anchorage, AK, 2003.

[17] H. Pishro-Nik, F. Fekri, Results on punctured LDPC codes, in: IEEE Inform. Theory Workshop, San Antonio, TX, 2004.

[18] S. Sesia, G. Caire, G. Vivier, Incremental redundancy hybrid ARQ schemes based on low-density parity-check codes, IEEE Trans. Commun. 52 (8) (2004) 1311–1321.

[19] J. Chen, A. Khisti, D. M. Malioutov, J. S. Yedidia, Distributed source coding using serially-concatenated-accumulate codes, in: IEEE Inform. Theory Workshop, San Antonio, TX, 2004.

[20] R. G. Gallager, Low-density parity-check codes, Cambridge MA: MIT Press (1963).

[21] F. R. Kschischang, B. J. Frey, H.-A. Loeliger, Factor graphs and the sum-product algorithm, IEEE Trans. Inform. Theory 47 (2) (2001) 498–519.

[22] M. Yang, W. E. Ryan, Y. Li, Design of efficiently encodable moderate-length high-rate irregular LDPC codes, IEEE Trans. Commun. 52 (4) (2004) 564–571.

[23] J. Li, K. R. Narayanan, C. N. Georghiades, Product accumulate codes: a class of codes with near-capacity performance and low decoding complexity., IEEE Trans. Inform. Theory 50 (1) (2004) 31–46.

[24] M. Isaka, M. P. C. Fossorier, High rate serially concatenated coding with extended Hamming codes, IEEE Commun. Lett. 9 (2) (2005) 160–162.

[25] L. Bahl, J. Cocke, F. Jelinek, J. Raviv, Optimal decoding of linear codes for minimizing symbol error, IEEE Trans. Inform. Theory 20 (2) (1974) 284–287.

[26] S.-Y. Chung, T. J. Richardson, R. L. Urbanke, Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation, IEEE Trans. Inform. Theory 47 (2) (2001) 657–670.

[27] A. Amraoui, LTHC: LdpcOpt (2001).
URL `http://lthcwww.epfl.ch/research/ldpcopt`

[28] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, R. L. Urbanke, Finite-length analysis of low-density parity-check codes on the binary erasure channel., IEEE Trans. Inform. Theory 48 (6) (2002) 1570–1579.

[29] T. Tian, C. Jones, J. D. Villasenor, R. D. Wesel, Construction of irregular LDPC codes with low error floors, in: Proc. IEEE International Conf. on Communications, Anchorage, AK, 2003.

[30] A. Ramamoorthy, R. D. Wesel, Construction of short block length irregular low-density parity-check codes, in: Proc. IEEE International Conf. on Communications, Paris, France, 2004.