Compression with Side Information Using Turbo Codes

Anne Aaron and Bernd Girod
Information Systems Laboratory
Stanford University, Stanford, CA 94305
amaaron@stanford.edu, *bgirod@stanford.edu*

Abstract

We show that turbo codes can come close to the Slepian-Wolf bound in lossless distributed source coding. In the asymmetric scenario considered, $\mathbf{X}$ and $\mathbf{Y}$ are statistically dependent signals and $\mathbf{X}$ is encoded with no knowledge of $\mathbf{Y}$. However, $\mathbf{Y}$ is known as side information at the decoder. We use a system based on turbo codes to send $\mathbf{X}$ at a rate close to $H(\boldsymbol{X}|\boldsymbol{Y})$. We apply our system to binary sequences and simulations show performance close to the information-theoretic limit. For distributed source coding of Gaussian sequences, our results show significant improvement over previous work. The scheme also performs well for joint source-channel coding.

1 Introduction

Consider a communication system with two statistically dependent signals, $\mathbf{X}$ and $\mathbf{Y}$ (Figure 1). Assume that $\mathbf{X}$ and $\mathbf{Y}$ come from two separate sources that cannot communicate with each other. The receiver, on the other hand, sees both encoded signals and performs joint decoding. Such a system requires *distributed source coding* since the encoders are distributed and the signals are compressed independently. An example of such a system is a wireless imaging network composed of low-complexity spatially separated nodes, sending correlated image information to a central processing unit. What is the minimum encoding rate required such that both $\mathbf{X}$ and $\mathbf{Y}$ can be recovered perfectly at the joint decoder?

If a joint encoder and a joint decoder are used, the minimum achievable combined rate for probability of decoding error to approach zero is simply the joint entropy $H(\mathbf{X}, \mathbf{Y})$. Surprisingly, as proven by Slepian and Wolf in 1973 [1], a combined rate of $H(\mathbf{X}, \mathbf{Y})$ is sufficient even if the correlated signals are encoded separately, but decoded jointly. According to the Slepian-Wolf coding theorem, the achievable rate region for distributed sources $\mathbf{X}$ and $\mathbf{Y}$ is given by

[0]This work is supported in part by a gift from Ericsson, Sweden and a Cisco Systems Stanford Graduate Fellowship.

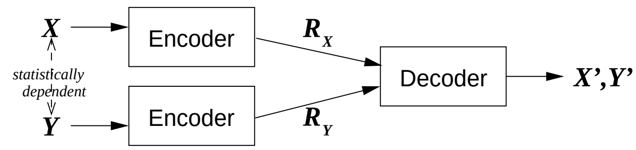

Figure 1: Distributed source coding. Statistically dependent signals $\mathbf{X}$ and $\mathbf{Y}$ are compressed independently and sent to a joint decoder.

$$\begin{aligned} R_{\mathbf{X}} + R_{\mathbf{Y}} &\geq H(\mathbf{X},\mathbf{Y}) \\ R_{\mathbf{X}} &\geq H(\mathbf{X}|\mathbf{Y}) \\ R_{\mathbf{Y}} &\geq H(\mathbf{Y}|\mathbf{X}) \end{aligned} \tag{1}$$

The Slepian-Wolf theorem suggests that it is possible to compress statistically dependent signals in a distributed manner to the same rate as with a system where the signals are compressed jointly.

The problem of designing practical codes which achieve the Slepian-Wolf bound is still open. Some work has been carried out [2-4] in designing codes but the performance is still not close to the information theoretic bound and the systems studied are applicable only to limited statistical models. In [2] and [3], the DISCUS (*Distributed Source Coding Using Syndromes*) method has been presented and applied to the asymmetric and symmetric scenarios. This method proposes that instead of sending the codeword representing $\mathbf{X}$, the syndrome of the codeword coset is sent and the receiver decodes by choosing the codeword in the given coset that is closest to $\mathbf{Y}$. DISCUS, with trellis and lattice encoding, was used to encode statistically dependent Gaussian sources. In [4] the authors used embedded trellis codes to encode Gaussian sources for the asymmetric case.

In this paper we work with the asymmetric distributed source coding scenario (Fig. 2) where we assume that $\mathbf{Y}$ is sent at the rate $R_{\mathbf{Y}} = H(\mathbf{Y})$ and is recovered perfectly at the decoder. We seek to send $\mathbf{X}$ close to the Slepian-Wolf bound of $R_{\mathbf{X}} = H(\mathbf{X}|\mathbf{Y})$ to achieve an overall rate $R_{\mathbf{X}} + R_{\mathbf{Y}}$ close to $H(\mathbf{X},\mathbf{Y})$. This asymmetric scenario is known as source coding $\mathbf{X}$ with *receiver side information* $\mathbf{Y}$.

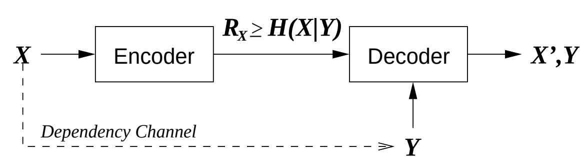

Figure 2: Source coding $\mathbf{X}$ with side information $\mathbf{Y}$. $\mathbf{Y}$ is sent at rate H($\mathbf{Y}$) and recovered perfectly at the decoder. $\mathbf{X}$ is sent at rate $\mathrm{R}_{\mathbf{X}} \geq \mathrm{H}(\mathbf{X}|\mathbf{Y})$

Lossless compression with receiver side information interestingly includes the case of lossless data compression, where the statistics of the data to be compressed are available only at the receiver. For this scenario, all the statistical information about $\mathbf{X}$ would be carried by a hypothetical $\mathbf{Y}$. The Slepian-Wolf theorem (1) suggests that optimal encoder-decoder pairs exist, where the encoder treats all source bits as equally likely and only the decoder takes the actual statistics into account. This is certainly very different from most source coding schemes, such as Huffman coding or arithmetic coding, where the coder is designed to take advantage of the source statistics.

We propose the use of turbo codes, a channel coding technique, to compress $\mathbf{X}$. Since $\mathbf{X}$ and $\mathbf{Y}$ are statistically dependent, the side information $\mathbf{Y}$ at the decoder can be viewed as a *channel output* where $\mathbf{X}$ is the input to the *dependency channel*. The only difference from simple channel coding is that there are additional bits sent, at rate $R_{\mathbf{X}}$, to the decoder from the encoder of $\mathbf{X}$ which are also used in the decoding. This dependency channel concept is the motivation for applying channel coding techniques to this source coding problem. Among the different channel coding methods, turbo codes, first introduced in 1993 by Berrou, et al. [5], have been demonstrated to possess superior error correction capabilities. These codes allow systems to perform very close to the Shannon channel capacity limit [5]. This is partially attributed to the "random" codewords produced by the interleaving in turbo coding.

In this paper we show how a system based on turbo codes can come close to the Slepian-Wolf performance bound in the asymmetric distributed source coding scenario. Garcia-Frias and Zhao independently developed a very similar technique for the symmetric case and have published their results in a recent letter [6].

The paper is organized as follows. In Section 2 we present the turbo coder and decoder structure. Section 3 discusses the simulation details and results. Section 4 describes how the system can be used for joint source-channel coding.

2 Turbo Coder and Decoder

2.1 Compression of Binary Sequences

Let $\mathbf{X}$ and $\mathbf{Y}$ be independent, identically distributed (i.i.d.) binary sequences $X_1X_2...X_L$ and $Y_1Y_2...Y_L$ with equally probable zeroes and ones. Let X_i be independent of all Y_j for $i \neq j$ but statistically dependent on Y_i. The $\mathbf{X}$-$\mathbf{Y}$ dependency is fully described by the conditional probability mass function $P(y|x)$.

In our asymmetric scenario, one of the signals ($\mathbf{Y}$) is compressed conventionally and sent at full rate $R_{\mathbf{Y}} \geq H(\mathbf{Y})$, such that it is available, without error, as side information for decoding $\mathbf{X}$ at the decoder. Very efficient techniques to achieve a rate $R_{\mathbf{Y}}$ very close to $H(\mathbf{Y})$ are widely known, hence we do not consider the compression of $\mathbf{Y}$ in detail.

We encode $\mathbf{X}$ as follows (Fig. 3): $\mathbf{X}$ is fed into a rate $\frac{n-1}{n}$ systematic convolutional encoder (for every input block of $n-1$ bits, this type of encoder produces codewords

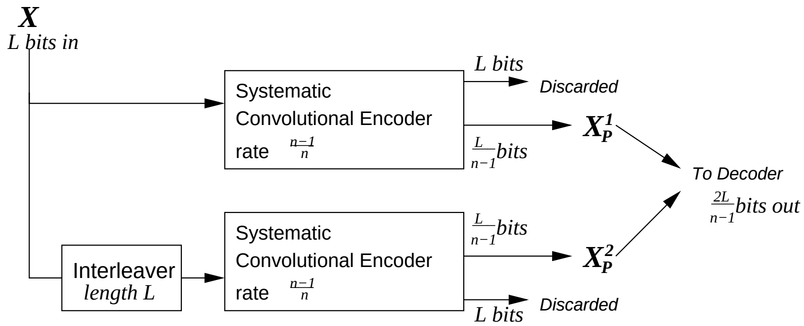

Figure 3: Encoder structure. $\mathbf{X}$ is encoded using two identical constituent convolutional encoders of rate $\frac{n-1}{n}$. The signal is interleaved before it is fed into the second convolutional encoder. Only the parity bits are sent to the decoder resulting in $R_{\mathbf{X}} = \frac{2}{n-1}$. Lower rates can be attained by puncturing the parity bits.

of length n composed of the actual input bits and one parity bit). In parallel, a randomly interleaved version of $\mathbf{X}$ is used as an input to another constituent convolutional encoder of the same type. At the output of the two encoders, we only send the parity bits, $\mathbf{X}_P^1$ and $\mathbf{X}_P^2$, of the codewords. This results in an encoding rate of $R_X = \frac{2}{n-1}$bit per input bit. We can also puncture the parity bits to achieve lower encoding rates.

Like standard turbo decoders, the decoder of our system is composed of two soft-input soft-output (SISO) constituent decoders (Fig. 4). In standard turbo decoding, a SISO block calculates extrinsic and *a posteriori* bit probabilities from the *a priori* probabilities and channel probabilities using the forward and backward recursion method. The only difference of our system is that the channel probabilities are calculated from the parity bits, $\mathbf{X}_P^1$ and $\mathbf{X}_P^2$, sent by the encoder and the *dependency channel* output $\mathbf{Y}$ which is available as side information at the decoder. The iterative decoding is executed by passing the extrinsic bit probability results of one SISO block as the *a priori* bit probabilities of the other SISO decoder. Iterations between the two constituent decoders are performed until a satisfactory convergence is reached. The final estimate $\mathbf{X}' = X_1'X_2'...X_L'$ is based on thresholding the *a posteriori* probabilities of the last iteration.

2.2 Extension to Continuous-Valued Sequences

The proposed system can be extended for source coding continuous-valued sequences. Let $\mathbf{X}$ be a sequence of i.i.d random variables $X_1X_2...X_L$ drawn from the probability distribution function $f(x)$. Let $\mathbf{Y}$ be a noisy version of $\mathbf{X}$ such that $Y_i = X_i + Z_i$, where Z_i is continuous, i.i.d and independent of X_i. The $\mathbf{X}$-$\mathbf{Y}$ statistical dependency can be described by the conditional probability distribution $f(y|x)$.

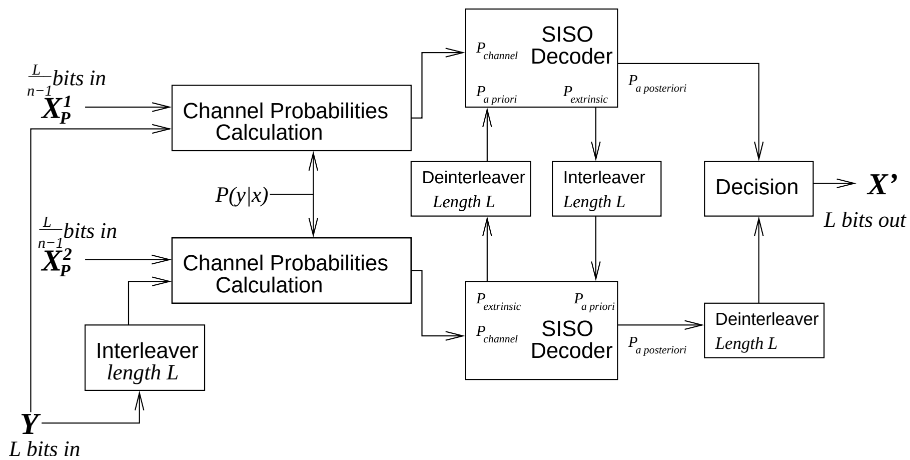

Figure 4: Decoder structure. The decoder uses the parity bits from the encoder, X_P^1 and X_P^2, and the side information Y to calculate the channel probabilities used by the SISO blocks. The two consituent SISO decoders exchange extrinsic probabilities in the usual iterative manner. When convergence is reached, the *a posteriori* probabilities of the last iteration is used to form the final estimate $\mathbf{X}'$.

Before the actual encoding (Fig. 5), we quantize each X_i using 2^M quantization levels producing a discrete sequence $\mathbf{X}_Q = X_{Q1}X_{Q2}...X_{QL}$ where $X_{Qi} \in \{u_1, u_2, ..., u_{2^M}\}$. We then convert the symbols into binary codewords generating a binary sequence $\mathbf{X}_B^1$ composed of ML bits. The sequence $\mathbf{X}_B^1$ is fed into the first constituent encoder. For the input to the second constituent encoder, the discrete sequence $\mathbf{X}_Q$ is first interleaved and then converted into a length ML binary sequence $\mathbf{X}_B^2$. Like the basic system, only the parity bits (or a punctured version of them) are sent to the decoder. Turbo decoding is performed similar to the basic system except that the channel probabilities are calculated using a continuous probability distribution function $f(y|x)$. Also, instead of propagating extrinsic bit probabilities, the SISO decoders propagate extrinsic symbol probabilities.

3 Simulation Results

We simulate and test the proposed system using binary and Gaussian random sequences. For all the simulations we use two 16-state, rate $\frac{4}{5}$ systematic convolutional encoders as the constituent encoders. These encoders have a generator matrix formed by the generator polynomials (shown in octal form) listed in Table 1. A total of 10^7 samples are used for each data point in the plots.

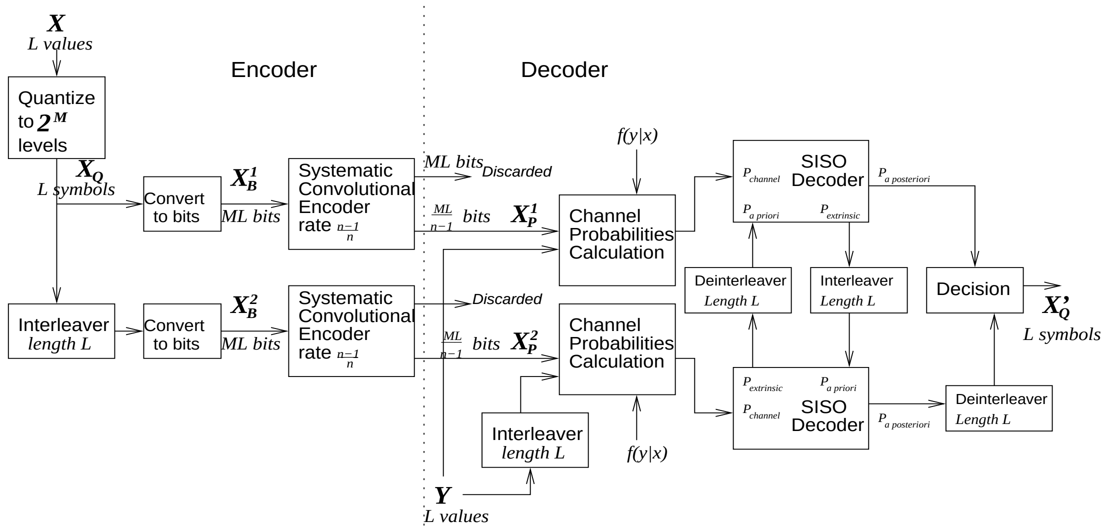

Figure 5: Encoder-Decoder structure for continuous-valued $\mathbf{X}$. $\mathbf{X}$ is quantized into symbols and converted into a binary sequence before encoding. At the turbo decoder, the conditional probability function $f(y|x)$ is used to calculate channel probabilities. The iterative decoding process propagates symbol probabilities. A final estimate $\mathbf{X}'_Q$ is formed using the *a posteriori* probabilites of the last iteration.

	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
$h_k(D)$	23	35	31	37	27

Table 1: Generator matrix of constituent encoders.

3.1 Binary Sequences

For our first set-up, we use binary sequences $\mathbf{X}$ and $\mathbf{Y}$ which have symmetric dependencies, that is, $P(X_i = Y_i) = 1-p$ and $P(X_i \neq Y_i) = p$, where p is the cross-over probability. We use sequence lengths of $L = 10^4, 10^5, 10^6$ and a maximum of 24 decoder iterations are allowed. Since $n = 5$ (we use a rate $\frac{4}{5}$ code) and no puncturing is involved, $R_X = 0.5$ bits per input bit. From the Slepian-Wolf theorem, we should theoretically be able to send $\mathbf{X}$ without error when $H(\mathbf{X}|\mathbf{Y}) \leq 0.5$. We vary p in our system, and therefore $H(\mathbf{X}|\mathbf{Y})$, and plot the probability of error (Fig. 6).

As it can be seen from the plot, the probability of error P_e has a very steep drop, similar to the ideal curves predicted by the Slepian-Wolf theorem. It can also be observed that the greater the sequence length L, the better the performance. This is consistent with the *interleaver gain* concept of turbo coding. One way to understand this is that the larger the interleaver, the more random the code-words generated, bringing the system closer to the random coding and asymptotic equipartition principles used in the Slepian-Wolf proof.

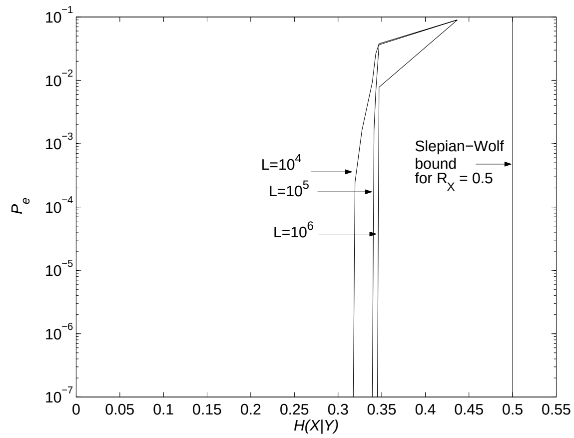

Figure 6: P_e plot for $R_X = 0.5$. Sequences of length L=10^4, 10^5, 10^6 were used. A total of 10^7 bits were simulated per sample point. P_e curves are steep and approach the Slepian Wolf bound.

We puncture the parity bits to achieve lower rates. The results for three different rates, using sequence length $L = 10^6$, are summarized in Table 2. The $H(\mathbf{X}|\mathbf{Y})$ value in the table is the point at which the probability of decoding error drops to less than 10^{-6} when rate $R_{\mathbf{X}}$ is used. As shown in the fourth column of the table, we have an inefficiency of about 35 - 45 % when $R_{\mathbf{X}}$ is compared to the ideal value $H(\mathbf{X}|\mathbf{Y})$. The table also shows the ratio of the combined rate $R_{\mathbf{X}} + H(\mathbf{Y})$ and the ideal combined rate $H(\mathbf{X},\mathbf{Y})$. This ratio decreases as the **X** and **Y** dependency increases. Assuming that **Y** is sent to the decoder at the ideal rate $H(\mathbf{Y})$, we have a combined rate inefficiency of 3 - 11 %. The last column compares our combined rate with the rate required if **X** and **Y** were to be encoded and decoded independently.

$R_{\mathbf{X}}$	$H(\mathbf{X}\|\mathbf{Y})$	$R_{\mathbf{X}} - H(\mathbf{X}\|\mathbf{Y})$	$\frac{R_{\mathbf{X}}}{H(\mathbf{X}\|\mathbf{Y})}$	$\frac{R_{\mathbf{X}}+H(\mathbf{Y})}{H(\mathbf{X},\mathbf{Y})}$	$\frac{R_{\mathbf{X}}+H(\mathbf{Y})}{H(\mathbf{X})+H(\mathbf{Y})}$
0.125	**0.089**	0.036	1.404	1.033	0.563
0.250	**0.185**	0.065	1.351	1.055	0.625
0.500	**0.346**	0.154	1.445	1.114	0.750

Table 2: Rate R_X achieved by our system as compared to the Slepian-Wolf bound $H(X|Y)$.

We compare our results with those from [6] where a scheme using turbo codes was used for the symmetric distributed source coding scenario. When there is high **X** and **Y** dependency, our asymmetric system performs closer to the Slepian-Wolf curve than the system in [6]. This is mainly due to our assumption that **Y** is sent at the ideal rate $H(\mathbf{Y})$. This suggests that it may be more efficient to use

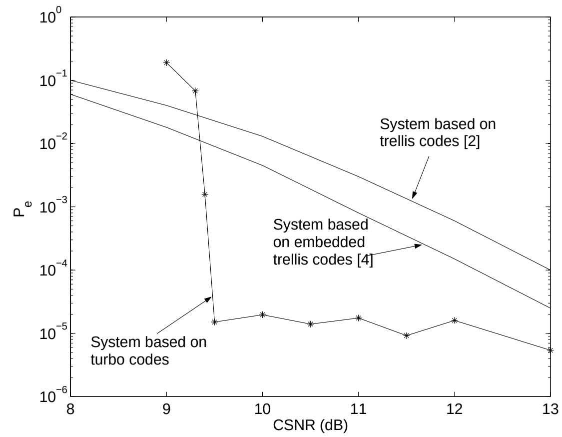

Figure 7: Performance of our system when Gaussian sources are used. P_e is plotted against the CSNR. Encoding rate is 1 bit per source sample. Our results are compared to the results in [2] and [4].

our asymmetric scheme for distributed source coding since there already exist very efficient techniques to encode at rate close to $H(\mathbf{Y})$.

3.2 Gaussian Sequences

In the second simulation set-up, we adopt the scenario used in [2] and [4] for comparison. Let $\mathbf{X}$ be a sequence of i.i.d Gaussian random variables $X_1X_2...X_L$ with mean zero and variance one. Let $\mathbf{Z}$ be a sequence of i.i.d Gaussian random variables $Z_1Z_2...Z_L$ with zero mean and variance σ^2 and independent of $\mathbf{X}$. We call the ratio of the variance of $\mathbf{X}$ and $\mathbf{Z}$ the *Correlation-SNR* (CSNR). $\mathbf{Y}$ is dependent on $\mathbf{X}$ such that $Y_i = X_i + Z_i$. Therefore, $f(y|x)$ is a Gaussian probability density function.

We apply the system described in Section 2.2 to encode and decode $\mathbf{X}$. Before actual encoding, $\mathbf{X}$ is quantized using a 4-level Lloyd-Max scalar quantizer resulting in an encoder input bit sequence of length $2L$. At the output of the constituent encoders no parity bit puncturing is performed, hence the resulting rate is 1 bit per source sample. Through the turbo decoding process, the decoder uses the side information $\mathbf{Y}$ and the parity bits from the encoder to form a symbol sequence estimate $\mathbf{X}'_Q$.

We vary the CSNR and plot the resulting probability of symbol error, P_e, which is $P(X_{Qi} \neq X'_{Qi})$ (Fig. 7). The simulations were performed with sequence length $L = 10^5$ and a total of 10^7 source samples per data point. Our results are compared with those in the work of Pradhan and Ramchandran [2] and Wang and Orchard [4]. In both papers, the authors also perform 4-level quantization and encode at a rate of 1 bit per source sample. As it can be seen from the plots, our scheme has about 3 dB improvement at $P_e = 10^{-4}$ over the embedded trellis code scheme proposed in [4]. At higher CSNR's, the P_e gap between our system and previous systems diminishes. The steep drop in probability of error, that information theory suggests, can be clearly seen in our results unlike the plots of previous work.

4 Joint Source-Channel Coding

Our system can easily be extended for joint source-channel coding. Assume that instead of arriving at the decoder with no error, the parity bit sequences $\mathbf{X}_P^1$ and $\mathbf{X}_P^2$ pass through a memoryless channel with capacity C and arrive at the decoder as $\mathbf{V}_P^1$ and $\mathbf{V}_P^2$, where $V_{P,i}$ is related to $X_{P,i}$ by the conditional probability $P(v_P|x_P)$. To account for the channel, the decoder simply includes the information $P(v_P|x_P)$ in its calculation of the SISO block input $P_{channel}$.

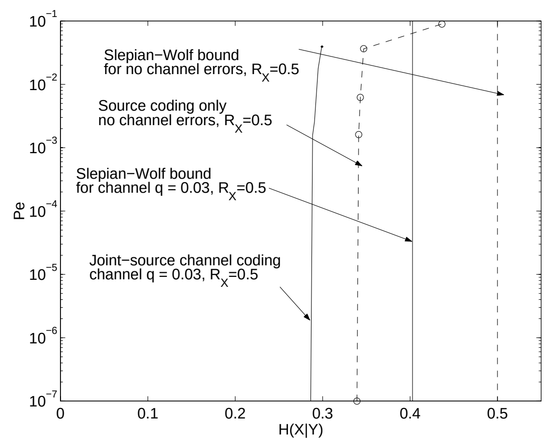

Figure 8: P_e when system is used for joint source-channel coding. Parity bits pass through a binary symmetric channel with cross-over probability $q = 0.03$ and channel capacity, $C = 0.80$. The encoder rate $R_{\mathbf{X}} = 0.5$. Also plotted are the results when $C = 1$ (source coding only).

We simulate the performance of our system used as a joint source-channel coder for binary sequences. Let the channel be a binary symmetric channel with cross-over probability q. Since the parity bits are sent through a channel with capacity $C(q)$, the effective transmission rate to the decoder is $R_{\mathbf{X},Decoder} = C(q)R_{\mathbf{X}}$. Based on the Slepian-Wolf theorem, we should be able to transmit $\mathbf{X}$ without error at an encoding rate $R_{\mathbf{X}} \geq \frac{H(\mathbf{X}|\mathbf{Y})}{C(q)}$. We vary the $\mathbf{X}$-$\mathbf{Y}$ dependency while keeping q constant and observe the resulting P_e. Plots for $q = 0.03$ and $q = 0$ (no channel error, system is used for source coding only) are shown in Fig. 8. The two q values yield similar looking curves, except that the curve for the joint source-channel coding case (and its corresponding ideal curve) is shifted to the left, with the shift based on the channel capacity C. For $q = 0.03$, the system had an $R_{\mathbf{X}}$ inefficiency of 40% and a combined rate inefficiency, $\frac{CR_X + H(Y)}{H(X,Y)} = 9\%$.

5 Conclusion

In this paper we proposed an encoder-decoder scheme based on turbo codes for asymmetric distributed source coding. The source signal $\mathbf{X}$ is encoded with no knowledge of $\mathbf{Y}$ using two constituent encoders. At the encoder output, only the parity bits, or a punctured subset, are sent to the decoder to achieve the desired rate. The decoder performs turbo decoding using the parity bits received and the side information $\mathbf{Y}$.

When applied to binary sequences, our system has a gap from the Slepian-Wolf bound ranging from 0.036 to 0.154 bits. This translates to a combined rate inefficiency of 3 to 11%. Our system was also applied to Gaussian sources and the results showed 3 dB improvement over previous work at $P_e = 10^{-4}$.

The proposed system can be used for joint source-channel coding. The decoder can simply incorporate both the channel statistics and the $\mathbf{X}$-$\mathbf{Y}$ dependency into the turbo decoding process.

References

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

[2] S.S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *Proc. IEEE Data Compression Conference*, Snowbird, UT, pp. 158 -167, March 1999.

[3] S.S. Pradhan and K. Ramchandran, "Distributed source coding: Symmetric rates and applications to sensor networks," *Proc. IEEE Data Compression Conference*, Snowbird, UT, pp. 363 -372, March 2000.

[4] X. Wang and M. Orchard, "Design of trellis codes for source coding with side information at the decoder," *Proc. IEEE Data Compression Conference*, Snowbird, UT, pp. 361 -370, March 2001.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-Codes," *Proc. IEEE International Conference on Communications*, Geneva, Switzerland, pp. 1064 -1070, May 1993.

[6] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Communication Letters*, vol. 5, no. 10, pp. 417 -419, October 2001.